



**Universitat Autònoma
de Barcelona**

Entorn web per a consultes on-line sobre bases de dades d'imatges

Memòria del projecte

d'Enginyeria Tècnica en

Informàtica de Sistemes

realitzat per

Carles Sánchez Ramos

i dirigit per

Ramon Baldrich Caselles

Escola Universitària d'Informàtica

Sabadell, Juny de 2009

El/la sotasignant, Ramon Baldrich Caselles ,
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Carles Sánchez Ramos
I per a que consti firma la present.
Sabadell, Juny de 2009

Signat: Ramon Baldrich Caselles

El/la sotasignant coodirector, Eduard Vázquez Marco ,
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Carles Sánchez Ramos
I per a que consti firma la present.
Sabadell, Juny de 2009

Signat: Eduard Vázquez Marco

Descripció

Aquest projecte es basa en la creació d'un entorn web. Aquest consisteix en un sistema per a fer consultes on-line a bases de dades d'imatges. Aquestes consultes es fan per contingut, a partir de mètodes creats per el grup de recerca CAT (color and texture) del CVC (Centre de Visió per Computador). L'aplicació web permet afegir nous mètodes que els membres del grup han programat amb matlab. Aquests són carregats automàticament per la web, construint els formularis associats de forma dinàmica. Això permet als membres del grup fer proves amb els mètodes i visualitzar els resultats de forma gràfica i on-line. A més també serveix com a mitjà de divulgació científica, mostrant a la resta del mon, demostracions de les investigacions que s'estan realitzant.

Agraïments

Al director Ramon Baldrich i al codirector Eduard Vàzquez.

A tots els amics i família pel seu suport incondicional , en especial a la Sònia París i el Miquel Calduch.

INDEX

1	Introducció.....	8
1.1	Situació.....	8
1.2	Motivacions.....	8
1.3	Objectius.....	9
1.4	Organització.....	10
2	Planificació.....	11
2.1	Estudi de mercat.....	11
2.2	Planificació.....	13
2.3	Estudi de Viabilitat.....	14
2.3.1	Viabilitat tècnica.....	15
2.3.2	Viabilitat econòmica.....	16
2.4	Requeriments funcionals i no funcionals	18
2.4.1	Actors.....	18
2.4.2	Casos d'us i especificacions.....	19
2.4.3	Diagrames de casos d'us.....	26
2.4.4	Diagrames de seqüència.....	28
2.5	Eines de desenvolupament.....	30
2.5.1	Sistema operatiu.....	30
2.5.2	Servidor web.....	30
2.5.3	Matlab a la web (cgi).....	32
2.5.4	Llenguatges de programació.....	34
2.5.5	Bases de dades.....	36
3	Desenvolupament.....	37
3.1	Metodologia de treball.....	37
3.2	Fonaments teòrics.....	37
3.2.1	Programari matlab via web.....	38
3.2.2	Api de flickr.....	39
3.2.3	Característiques dels mètodes matlab.....	40
3.2.4	Programació.....	41

3.3	Anàlisi.....	42
3.3.1	Organització de la web.....	42
3.3.2	funció de cada component.....	44
3.3.3	Diagrama web-components.....	46
4	Disseny i implementació.....	47
4.1	Fases.....	47
4.1.1	Muntatge del servidor web.....	47
4.1.2	Configuració dels components.....	49
4.1.3	Creació de la web.....	54
4.1.3.1	Estructura de la web.....	55
4.1.3.2	Consulta de mètodes.....	59
4.1.3.3	Execució de mètodes.....	64
4.1.3.4	Cercador d'imatges flickr.....	68
4.1.3.5	Configuració web.....	72
4.1.3.6	Control d'errors.....	76
4.1.3.7	Fragments de codi a destacar.....	77
4.1.4	Aplicació d'un estil.....	82
4.2	Proves d'integració.....	91
5	Conclusions.....	95
5.1	Valoració del projecte.....	95
5.2	Conclusions del projecte.....	96
5.3	Treball futur.....	97
6	Bibliografia.....	99

1. INTRODUCCIÓ

1.1. Situació

Internet a esdevingut el mitjà de divulgació per excel·lència. Això pot ser molt beneficiós en la cooperativitat entre grups de recerca dedicats a camps relacionats ja que aquests poden compartir coneixements i resultats i estar informats sempre dels darrers avenços. És doncs important que els investigadors puguin ensenyar les seves vies de recerca i quins són els resultats obtinguts.

Aquest projecte, emmarcat doncs en el context de la divulgació científica, es centre en el grup color and texture (CAT) del Centre Visió per Computador (CVC) de la Universitat Autònoma de Barcelona (UAB). Aquest grup treballa amb mètodes per a descriure les característiques del color i la textura. La majoria d'aquests mètodes es posen a prova i es codifiquen en matlab. Per fer aquestes proves es treballa amb bases de dades d'imatges i s'extreuen uns resultats.

Aquest projecte es centrarà en buscar alguna solució tan per donar a conèixer mètodes investigats com també per posar-los a prova.

1.2. Motivacions

La indiscutible importància d'Internet esdevé, doncs, una motivació essencial en el present projecte. La mateixa idea d'utilitzar un mitjà a l'abast de tothom, motiva, per coherència, l'utilització de software lliure. Això també propicia l'ús de funcionalitats ja implementades i provades, amb un baix cost relatiu a la seva adaptació.

D'altre banda i fent referència al punt anterior el grup CAT (com molts grups de recerca en ciència) es creen els seus propis programes en matlab per treballar després sobre imatges buscant les característiques desitjades i provar els mètodes, d'entre aquests mètodes destaquen aquells que busquen la semblança entre imatges . Un cop s'aconsegueixen els resultats esperats el mètode és vàlid, i és en aquest punt en que és té la necessitat de mostrar el que s'ha descobert, a tota la comunitat. Per tan aquí

entra la segona gran raó del projecte, poder fer demostracions dels mètodes investigats, i la manera de fer aquestes demostracions és mitjançant la web.

1.3. Objectius

L'ús d'Internet com a mitjà de divulgació científica marca els objectius d'aquest projecte. Així doncs, es vol crear una *interface* en el qual es puguin controlar mètodes d'anàlisi d'imatges des d'una pàgina web, facilitant així la seva divulgació.

Per una banda la pàgina haurà de tenir una part pública perquè qualsevol visitant pugui utilitzar els mètodes disponibles (principalment altres investigadors en el mateix camp). En aquesta part es podran provar un conjunt de mètodes seleccionats sobre un conjunt d'imatges (en l'apartat d'especificacions és detallaran les funcionalitats concretes). Després es mostrarà la sortida del mètode perquè es pugui comprovar el seu funcionament. Els mètodes que es faran servir pel prototip del projecte, seran mètodes que busquen la semblança entre imatges.

La web també inclourà una part d'intranet en la que els usuaris del grup podran administrar el *site* i crear les seves pròpies bases de dades d'imatges descarregant-les (amb prèvia consulta) directament de la base de dades de flickr amb milions d'imatges i on cada dia n'hi han més.

Per últim i per posar a prova la funcionalitat del projecte, la web s'integrarà amb un altre projecte realitzat a la ETSE (Escola tècnica superior d'enginyeria). Aquest projecte tracta justament sobre fer uns mètodes concrets per al processament d'imatges. L'objectiu serà doncs demostrar que l'aplicació web es capaç d'integrar mètodes programats en matlab fent la integració inicial amb aquest altre projecte. Caldrà doncs concretar els requeriments per al final connectar els dos projectes.

El manteniment de la web no serà realitzat per un expert. Així doncs, uns altres objectius seran la flexibilitat del projecte (introducció/manteniment de mètodes amb facilitat) i la seva portabilitat (el més independent possible d'altres programes).

El idioma inicial de la web s'adaptarà a les necessitats del grup de recerca i de la comunitat científica. Per tan, doncs, la web serà en anglès.

1.4. Organització

El projecte realitzat s'organitza en 4 capítols els quals s'expliquen a continuació:

➤ **Planificació**

Es descriu la part pre-projecte. En aquest capítol es fa l'anàlisi sobre com ha de ser la planificació, quines aplicacions semblants existeixen o quina és la viabilitat del projecte. També es fa referència a les eines que s'utilitzaran i els requeriments funcionals i no funcionals tenint en compte els usuaris que pot tenir i ho poden utilitzar. Cal dir que a mida que el projecte ha anat avançant els requeriments s'han anat refinant, aquí es mostraran els requeriments finals.

➤ **Desenvolupament**

És el capítol en el que podem trobar una explicació de quin ha sigut el procés de desenvolupament. Quins estudis s'han fet per assolir tots els fonaments teòrics. Tot l'anàlisi corresponent al flux d'informació entre aplicació i components, utilitzant els diagrames necessaris.

➤ **Disseny i implementació**

Descripció detallada de cada una de les fases en el disseny de tota l'aplicació web final i explicació de quin ha sigut el procés d'integració i testeig del *software*.

➤ **Conclusions**

En aquest darrer capítol de la memòria del projecte s'hi exposa una valoració del treball fet, una comparació de resultats amb els objectius inicials i per últim una reflexió de les possibles futures millores i funcionalitats.

➤ **Annexes**

Manual d'usuari de com s'utilitza l'aplicació i manual per administradors. També s'hi podrà trobar tota la codificació comentada de la web.


2. PLANIFICACIÓ

2.1. Estudi de mercat

Abans de començar a treballar amb l'estudi de viabilitat o començar a pensar i parlar sobre els requeriments de l'aplicació, cal fer una mirada a quins tipus d'aplicacions podem trobar (ja fetes) que tinguin característiques semblants al projecte que es vol realitzar. Ja sigui per extreure possibles idees, millorar aspectes o també per saber el grau competència que es podrà tenir.

Per aquesta raó ens posem a fer un estudi de la situació del mercat. Els objectius de la cerca són els següents:

- **Pàgines web que integrin programari realitzat amb matlab o octave.** Els programes científics més coneguts, el primer amb necessitat de llicència per a la seva utilització i el segon d'utilització lliure. El projecte s'enfocarà només a programari matlab. Els resultats a les cerques de webs que integrin aquest tipus de funcionalitat són escassos per no dir nuls ja que no s'ha trobat cap exemple.
- **Pàgines d'altres grups de recerca amb demostracions dels seus estudis.** Les cerques s'han fet sobre grups de recerca del mateix àmbit (treball en la visió per computador). La majoria d'aquests grups mostren els resultats obtinguts a les seves investigacions per mitjà d'esquemes, publicacions o vídeos explicatius. En escasses ocasions es compte amb demostracions *online* de la feina presentada. En el millor dels casos hi ha l'opció de baixar-se un codi Matlab.
- **Aplicacions o pàgines web que utilitzin les imatges de la base de dades flickr.** Veure quines funcionalitats se li pot donar aquesta base de dades veient exemples de utilitats fetes. Aquestes són algunes:

Logotip	Descripció
	Zflickr us senzill, atractiu visualment i amb una gran funcionalitat. Es fa la consulta sobre el que es vol buscar i es mostren els resultats en forma de mosaic. Al pitjar sobre una imatge es veu en gran i també la podem arrossegar cap al directori que vulguem.
	Pixolu cercador d'imatges a flickr però amb una interessant forma de buscar refinament. Podent utilitzar imatges trobades com a filtre per noves consultes. Més informació a la seva web http://www.pixolu.de/#
	Flickr services diferents aplicacions fetes per terceres persones per treballar amb les imatges emmagatzemades a flickr. <i>Es poden trobar totes a la web</i> http://www.flickr.com/services/

Taula 1. Cercadors a flickr.

- **Cercadors d'imatges existents** per tenir exemples de com buscar i mostrar les imatges en el particular cercador (d'imatges flickr) del projecte. S'han examinat entre d'altres, cercadors com:

Cercador		
		
		

Taula 2. Cercadors més coneguts.

Tots ells comparteixen la fàcil utilitat amb resultats ràpids i força acurats depenent de la cerca. La forma de presentar les imatges es molt semblant (forma de mosaic i amb un paginat). Tots ells també incorporen a les galeries d'imatges el nou component visor de fotografies per als navegadors més utilitzats (mozilla, explorer, safari) anomenat *cooliris*. Que permet veure i trobar la imatge cercada d'una forma molt més còmode.

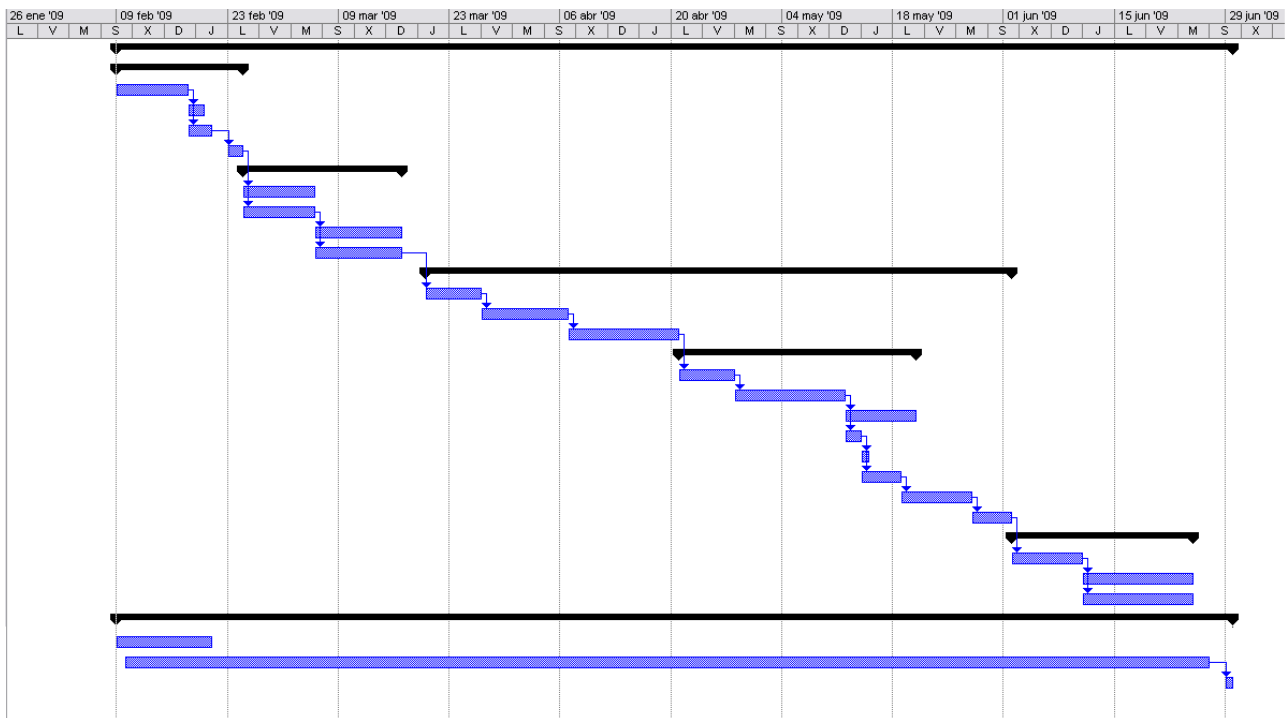
2.2. Planificació

El projecte serà realitzat per una persona (Carles Sánchez Ramos) el qual dedicarà entre 1 i 8 hores diàries els dies lectius.

La planificació del projecte esta dividia en sis punts generals: planificació amb un anàlisi dels requeriments (aquest han anar variant a mida que el projecte ha anat avançant). Desenvolupament , disseny i implementació de la web, integració de la web amb tots els components tan externs com interns i proves de funcionament. També el desenvolupament de la memòria com les reunions amb els tutors i l'altre projectista formen part de la planificació.

Per cada una d'aquestes etapes hi haurà el seu document associat. Tota la planificació completa ve definida en el següent diagrama de Gant:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	- Projecte	101 días	lun 09/02/09	lun 29/06/09	
2	- Planificació	12 días	lun 09/02/09	mar 24/02/09	
3	Anàlisi de requeriments	7 días	lun 09/02/09	mar 17/02/09	
4	Estudi de mercat	2 días	mié 18/02/09	jue 19/02/09	3
5	Estudi de viabilitat	3 días	mié 18/02/09	vie 20/02/09	3
6	Estudi de les eines de desenvolupament	2 días	lun 23/02/09	mar 24/02/09	5
7	- Desenvolupament	14 días	mié 25/02/09	lun 16/03/09	
8	Aprenentatge del funcionament de les eines	7 días	mié 25/02/09	jue 05/03/09	6
9	Característiques dels mètodes matlab	7 días	mié 25/02/09	jue 05/03/09	6
10	Anàlisi de la web	7 días	vie 06/03/09	lun 16/03/09	9
11	Anàlisi dels components de la web	7 días	vie 06/03/09	lun 16/03/09	9
12	- Disseny i implementació	52 días	vie 20/03/09	lun 01/06/09	
13	Muntatge del servidor	5 días	vie 20/03/09	jue 26/03/09	11
14	Configuració de components	7 días	vie 27/03/09	lun 06/04/09	13
15	Execució de programes matlab	10 días	mar 07/04/09	lun 20/04/09	14
16	- Creació de l'interfície web	22 días	mar 21/04/09	mié 20/05/09	
17	Creació dinàmica de formularis	5 días	mar 21/04/09	lun 27/04/09	15
18	Enviament i rebuda de dades matlab	10 días	mar 28/04/09	lun 11/05/09	17
19	Interfície de la resposta	7 días	mar 12/05/09	mié 20/05/09	18
20	Interfície de configuració	2 días	mar 12/05/09	mié 13/05/09	18
21	Pantalles d'informació (errors, carregan...)	1 día	jue 14/05/09	jue 14/05/09	20
22	Unió de tots els mòduls	3 días	jue 14/05/09	lun 18/05/09	20
23	Creació interfície del buscador a flickr	7 días	mar 19/05/09	mié 27/05/09	22
24	Aplicació d'un estil a la web	3 días	jue 28/05/09	lun 01/06/09	23
25	- Integració	17 días	mar 02/06/09	mié 24/06/09	
26	Unió entre els dos projectes (EUIS , ETSE)	7 días	mar 02/06/09	mié 10/06/09	24
27	Proves de funcionament	10 días	jue 11/06/09	mié 24/06/09	26
28	Correcció d'errors	10 días	jue 11/06/09	mié 24/06/09	26
29	- Documentació	101 días	lun 09/02/09	lun 29/06/09	
30	Reunions amb el tutor	10 días	lun 09/02/09	vie 20/02/09	
31	Memòria	99 días	mar 10/02/09	vie 26/06/09	
32	Manuale d'usuari	1 día	lun 29/06/09	lun 29/06/09	31

Figura 1. Planificació feta amb MS Project.*Figura 2. Diagrama de Gant fet amb el MS Project.*

2.3. Estudi de viabilitat

És el estudi inicial per determinar si aquest projecte es podrà fer o no. Per fer aquest estudi s'ha analitzat la viabilitat tècnica, operativa, econòmica i legal per finalment extreure unes conclusions en les que s'ha pogut saber si el projecte seria viable. Inicialment es comenten els avantatges i inconvenients de fer aquesta aplicació web.

➤ Inconvenients

- Utilització de tecnologies desconegudes pel realitzador del projecte, com per exemple: la integració de programes Matlab a la web, utilització de les imatges de flickr, programació en llenguatges desconeguts.
- Terminis per la implementació i implantació.
- Integració amb un altre projecte de la ETSE. El que suposa una gran coordinació entre els dos treballs per aconseguir la correcta integració.
- La unió dels diferents mòduls del projecte programats en diferents llenguatges serà un inconvenient a l'hora de depurar errors.

➤ **Avantatges**

- Un servei que donarà molta agilitat als investigadors per donar a conèixer els seus estudis.
- Fer un prototip web totalment flexible en qual afegir noves demostracions de mètodes investigats serà de gran facilitat i per tan podrà ser estès a altres grups d'investigació.
- No existeixen preus de llicències de programes externs ja que es realitza amb software lliure.
- No s'hauria de formar a cap treballador. N'hi haurà prou amb un manual de com introduir nous mètodes o canviar-ne d'existents.
- Menys temps perdut per als investigadors en crear les seves bases de dades d'imatges.
- Alt grau de portabilitat de mètodes (matlab) entre webs que utilitzin aquest projecte.

2.3.1. Viabilitat tècnica

Els majors riscos del projecte estan en la primera fase. És la fase en que s'han d'entendre les eines i mòduls amb les que es treballaran. Es necessita saber tot el possible per no cometre errors que ens facin tornar aquesta fase de nou. En l'apartat, '*eines de desenvolupament*', de la memòria es fa una explicació més detallada de quines han sigut les alternatives a escollir i quines han sigut les eines seleccionades per la seva viabilitat tan en el seu us particular com també en el conjunt del projecte. Si en aquesta fase ens adonéssim (no és el cas) que per exemple , executar codi matlab des de web, fer consultes o descarregar-se imatges de la base de dades de flickr, no és possible el projecte seria inviable tècnicament parlant.

També s'ha de tenir en compte que tot el que es faci ha d'estar enfocat perquè sigui una aplicació flexible i portable sense que els seus usuaris hagin d'aprendre cap tipus de llenguatge de programació per poder crear els seus propis formularis web que consultaran els seus mètodes.

Un altre punt conflictiu, com ja s'ha comentat en els inconvenients serà la integració de la web amb tots els seus mòduls. S'haurà de reservar força temps per poder fer

aquesta etapa sense que ens retrasi la data d'entrega. Serà complicat depurar els errors en el pas de flux de dades entre diferents softwares.

Com a punt positiu però cal mencionar que si tots els mòduls estan programats amb els requeriments demanats serà qüestió de temps i paciència el fet de tindre tota l'aplicació integrada.

Per tan el projecte seria tècnicament viable ja que es tenen a l'abast totes les eines necessàries per realitzar-l'ho tenint en compte uns requeriments (definits en el següent capítol) i el temps d'entrega.

2.3.2. Viabilitat econòmica

Els beneficis d'aquesta aplicació són grans pels investigadors que podran ensenyar al mon quins són els seus estudis demostrant els resultats obtinguts, o fer grans processos de prova dels seus mètodes visualitzant de forma gràfica els resultats obtinguts.

També hi han grans beneficis pels creadors del programa ja que es podrà oferir el mateix servei a altres centres d'investigació.

I finalment els costos són baixos (utilització de software lliure), la web serà fàcil d'utilitzar i administrar i només s'hauran de pagar les hores que s'hagin treballat per crear-la.

A continuació es mostra un exemple del pressupost del que costaria integrar aquesta aplicació web (demostració de mètodes) a la web d'algun altre grup de recerca dedicat al processament d'imatges:

Pressupost

CAT (CVC)

93 111 11 11

Nº PRESSUPOST 1

A/A: Color and texture group (Centre de Visió per Computador)

10/01/2009

Aplicació web per la demostració i prova de mètodes investigats en processament d'imatges utilitzant bases de dades d'imatges pròpies o amb la possibilitat de crear-te-les mitjançant un cercador a imatges flickr.

El prototip constarà de:

- Creació de bases de dades d'imatges descarregades de flickr
- Creació dinàmica de formularis dels mètodes
- Llançament des de web processos matlab
- Visualització de resultats per el mètode i imatges seleccionades.
- Intranet per administrar l'aplicació web.

Inclou :

Tots els programes necessaris per el funcionament de l'aplicació.

Hores treballades per part de tot l'equip (aprox. 400 h a 7€/h).

Manuais per a la creació de nous formularis per a nous mètodes.

No inclou :

Ampliacions i millores futures.

Manteniment del servidor web.

PVP :2.800 €

Termini de lliurament

1 Mes al firmar

Condicions de pagament :

A la acceptació del Pressupost:

40 %.

A la Finalització:

60 %.

*Aquests preus no inclouen IVA.**

Acceptació de Pressupost i condicions de pagament.

Aquesta oferta te una vigència de 30 dies.

2.3.3. Conclusió

Com a conclusió podem dir que aquest projecte és viable ja que, els costos que generarà seran baixos i els beneficis alts. Aquests ens permetran donar-nos a conèixer i als grups d'investigació oferir una millor coneixença del seu treball. A més com ja s'ha dit el que es construeix és una aplicació bàsica, per tan tindrà moltes acotacions que es podran anar ampliant lo que donarà uns beneficis de manteniment i millora de cara a nosaltres, però també millor servei a tots els usuaris que ho utilitzin.

2.4. Requeriments funcionals i no funcionals

Per fer un anàlisi dels requeriments de l'aplicació necessitem abans fer un resum de quins són els actors i casos d'us.

2.4.1. Actors

Són totes les persones i/o aparells que interactuen o estan d'alguna manera relacionats amb l'aplicació:

- **Usuaris d'internet** , utilitza la web per veure demostracions dels mètodes que més li interessin utilitzant les bases de dades d'imatges que estiguin disponibles. Aquest usuari només podrà accedir a la part pública de la web.
- **Usuaris administradors** ,utilitza la web per provar mètodes realitzats per ell o altres dels seus companys utilitzant les bases de dades d'imatges que li interessin. Aquest usuari pot veure tan la part pública com la part privada en la qual pot administrar la configuració de la web i també descarregar-se nous conjunts d'imatges (per després ser provats amb els mètodes) de la base de dades de flickr de manera online.
- **Matlab MCR (versió reduïda del programa)**, extern a l'aplicació executarà els processos (prèviament compilats) que se li diguin des de la web i ens retornarà uns resultats que els processos generen i que s'hauran de visualitzar. És el motor que permet l'execució de codi matlab compilat sense tindre el matlab instal·lat.

- **Servidor**, lloc d'emmagatzemament de la informació. En aquest actor s'emmagatzemaran les imatges que es descarreguin de flickr.

2.4.2. Casos d'us i especificacions

En aquesta secció es nomenaran tots els casos d'us fent referència a quines són les especificacions per a cada un d'ells. Al final es faran dos diagrames de casos d'us (part privada de la web, part pública).

➤ **Consulta d'un mètode sobre un conjunt d'imatges**

Procés en el qual un usuari de la web prepara l'execució d'un mètode sobre un conjunt d'imatges. Tan el conjunt d'imatges com el mètode se seleccionen prèviament. Un cop seleccionats els dos paràmetres es genera el formulari corresponent al mètode seleccionat amb els paràmetres per defecte ja introduïts. És doncs el mòdul encarregat de recollir tots els paràmetres necessaris per a l'enviament de les dades a excussió.

Actors

Usuaris web.

flux d'events

Bàsic

- 1.-L'usuari entra a l'opció de menú de processar mètode.
- 2.-L'usuari escull quins directoris d'imatges vol processar i sobre quin mètode.
- 3.-El sistema valida les dades
- 4.-El sistema llegeix el arxiu del mètode.
- 5.-El sistema construeix el formulari (definit l'arxiu) amb els valors per defecte.

Alternatiu

- En el pas 3 si l'usuari no introdueix el conjunt d'imatges el sistema senyala el error i es torna al pas 2.
- En el pas 5, si ja s'ha fet una consulta anteriorment es carregaran les dades consultades.

Pre condicions

- Per entrar aquest mòdul només fa falta seleccionar la opció de menú.
- El programa no funcionarà si la configuració de la web no es correcte.

Post condicions

- Es guarda la ruta de les imatges seleccionades.
- Es guarda el mètode a utilitzar i les rutes de configuració de la web.

Requeriments no funcionals

- RNF1 , Les imatges han d'estar guardades al servidor dins del directori d'imatges, distribuïdes de tal manera que el nom del directori resumeixi el tipus d'imatges que es poden trobar al seu interior.

➤ **Execució d'un mètode**

Procés en el qual un usuari omple les dades del formulari del mètode i l'executa. L'usuari pot executar directament el mètode sense toca cap paràmetre ja que inicialment apareix amb els camps per defecte omplerts.

Actors

Usuaris web.

flux d'events

Bàsic

- 1.-L'usuari tria una imatge per pujar-la al servidor (és la imatge amb la que ell vol consultar) la qual s'utilitzarà per ser comparada amb el conjunt d'imatges escollit en el anterior mòdul.
- 2.-El sistema carrega la imatges i torna a mostrar el formulari amb la imatge escollida.
- 3.-Tornem al pas 1 o continuem al pas 4
- 4.-L'usuari canvia les dades que necessiti o les deixa tal com estan.
- 5.-L'usuari executa el procés del mètode.

Alternatiu

- En el pas 1 si l'usuari no tria cap imatge no serà un error. La comparació entre imatges es farà totes amb totes del conjunt seleccionat en el mòdul anterior.

Pre condicions

- Hi ha d'haver un conjunt d'imatges seleccionat prèviament.
- Si prèviament ja s'ha fet una consulta es carregarà de nou la o les imatges (que ja han sigut pujades al servidor) amb les que s'ha consultat en l'anterior excussió.

Post condicions

-Totes les dades que s'envien al mètode compilat han de ser dins de camps del formulari.

-Es crida a la *interface* (codi matlab) amb tots els paràmetres del mètode.

Requeriments no funcionals

-RNF1 , No hi haurà un control del tipus de dades que es puguin canviar en el formulari del mètode ja que si només es vol provar, les dades per defecte són les correctes.

➤ **Visualització dels resultats de l'execució d'un mètode**

Visualització per pantalla de les imatges ordenades de més a menys semblança amb les imatge/s pujades al servidor. Si no s'ha introduït imatge/s per comparar es mostra una gràfica i un text explicatiu de les estadístiques. Es visualitza en forma de paginat i amb un número d'imatges per pàgina que pot ser definit per el usuari.

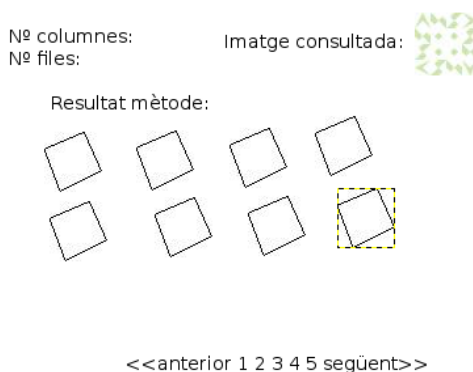


Figura 3. Aproximació del resultat

Actors

MCR (versió reduïda del matlab).

flux d'eventsBàsic

- 1.-Les dades que venent del formulari del mètode arriben a la capa més externa dels mètodes compilats.
- 2.-En aquesta capa *interfície* es cridarà els mètodes amb els seus paràmetres, per tan es preparen les crides als mètodes.
- 3.-Es crida al mètode amb els paràmetres necessaris. Entre ells l'adreça del arxiu amb el conjunt d'imatges i les variables de configuració.

- 4.-Es guarden tots els paràmetres per tornar a ser enviats cap a la web.
- 5.-S'executa el mètode de.
- 6.-Es crida a la pàgina de resposta (on es mostraran els resultats), passant-li els paràmetres necessaris.
- 7.-Es creen els *thumbnails* de les imatges que s'han de mostrar.
- 8.-L'aplicació mostra la graella d'imatges.

Alternatiu

- En cas d'error en qualsevol dels punts l'aplicació ens enviarà a una pàgina d'error.
- Al pas 7 només es crearan els thumbnails de les imatges que no el tinguin.

Pre condicions

- Les dades obligatòries han d'existir (fitxer imatges, variables de configuració, fitxer de sortida d'imatges, mètode).

Post condicions

- S'ha tingut de crear un arxiu (d'adreces) amb el nou ordre del conjunt d'imatges escollit inicialment o amb un escrit descriptiu.

Requeriments no funcionals

- RNF1 , La graella de mostrar les imatges serà per defecte de 3x5.
- RNF2, Al clicar sobre una imatge es veurà la imatge a tamany real.
- RNF3, Només es visualitzaran imatges amb format: jpg,gif,png,jpeg. Les imatges .bmp es descartaran perquè són massa pesades per a treballar amb matlab.

➤ **Validació d'un usuari**

Procés en el qual els usuaris administradors (membres del grup) de la web s'identifiquen per poder accedir a la part privada (intranet).

Actors

Usuaris administradors.

flux d'events

Bàsic

- 1.-Els usuaris seleccionen entrar a una zona protegida de la web.
- 2.-Es demana nom d'usuari i password.
- 3.-Es validen les dades.
- 4.-Les dades són acceptades.

5.-Es dóna accés a l'intranet.

6.-Es mantenen les dades durant tota la sessió.

Alternatiu

-Al pas 5 si les dades són denegades no es dóna accés, es torna al menú inicial de la web (pas 1).

➤ **Configuració de la web**

Mòdul de la web encarregat de mantenir les dades de configuració del site. Aquestes dades es podran modificar a través de la intranet de la web. Aquest mòdul ha d'estar protegit ja que són configuracions que no poden ser modificades per usuaris d'internet.

Actors

Usuaris administradors.

flux d'events

Bàsic

1.-L'usuari administrador entra a la configuració de la web.

2.-Es visualitzen les dades guardades en aquell moment.

3.-L'administrador pot modificar qualsevol dada.

4.-L'administrador guarda les modificacions.

5.-Es mostra un missatge que les dades han sigut guardades correctament.

Alternatiu

-Al pas 3 l'administrador pot no modificar res i només consultar alguna dada i seguidament es torna a la intranet.

-Al pas 4 si hi ha alguna configuració buida es mostra un missatge d'error i es tornar al pas 3.

Pre condicions

-L'usuari administrador s'haurà d'haver validat.

Requeriments no funcionals

-RNF1 , Quan un administrador canvia alguna configuració ha d'estar segur dels nous valors que configura, ja que el funcionament de la web depèn d'aquests paràmetres.

➤ **Consulta d'imatges a flickr**

En aquest mòdul es gestionen les consultes a la base de dades d'imatges de flickr. A partir d'un filtratge s'acota el resultat de la cerca.

Actors

Usuaris administradors.

flux d'events

Bàsic

- 1.-S'entra a l'opció de consultes a flickr.
- 2.-L'usuari omple els camps pels que vulgui acotar la consulta.
- 3.-L'usuari executa la consulta.

Pre condicions

-L'usuari administrador s'haurà d'haver validat.

Alternatiu

-Al pas 2 si l'usuari no omple cap camp la consulta serà aleatòria.

Requeriments no funcionals

-RNF1 , Les paraules clau de la cerca amb un (-) davant significaran que es volen excloure de la consulta.

➤ **Visualització de resultats imatges flickr**

Visualització per pantalla de les imatges trobades a flickr que s'adaptin a les paraules clau consultades. Les imatges es presenten en format paginat i de més a menys importància amb relació a la cerca.

Actors

Usuaris administradors.

flux d'events

Bàsic

- 1.-Les imatges es presenten en la pantalla.
- 2.-L'usuari pot navegar per les pàgines o seleccionar alguna imatge per veure-la en gran.
- 3.-L'usuari pot tornar a fer una consulta o descarregar-se algunes imatges.

Alternatiu

- En el pas 1 si no hi han imatges resultants es mostra que no s'ha trobat res.
- En el pas 2 es mostrarà un error si al intentar veure la imatge en gran aquesta no existeix.

Requeriments no funcionals

-RNF1 , Les imatges es mostraran en una graella totes de la mateixa mida i quadrades. La graella dependrà del número d'imatges a mostrar.

➤ **Descàrrega d'imatges flickr**

Procés en el qual un usuari administrador es descarrega de flickr un conjunt d'imatges. Aquestes imatges són guardades a un directori (amb el nom que l'usuari vulgui) al servidor.

Actors

Usuaris administradors.

flux d'events

Bàsic

- 1.-L'usuari pot seleccionar a descarregar-se la pàgina sencera d'imatges que esta visualitzant o descarregar-se un número d'elles.
- 2.-Es demana quin és el nom del directori on es volen guardar les imatges.
- 3.-L'usuari executa la descàrrega d'imatges.
- 4.-Un cop les imatges s'han descarregat es mostra un missatge informatiu.
- 5.-L'usuari pot tornar a la consulta o a consultar de nou.

Alternatiu

- En el pas1 si l'usuari tria la opció de descarregar-se un número concret d'imatges de la pàgina haurà de donar l'interval de les posicions en que es troben les imatges inicial i final.
- Al pas 2 si el nom de directori ja existeix es guardaran en un directori amb el nom escollit seguit de la data en que es va fer la consulta.
- En el pas 4, mentres es descarreguen les imatges es restringirà a que només es descarregui imatges de mida gran, tenint en compte que a la base de dades de flickr tenen varies mides per cada imatge.

Pre condicions

-Totes les direccions de les imatges trobades a flickr han d'estar guardades.

Post condicions

-L'aplicació web tanca la connexió amb flickr.

Com a requeriments no funcionals generals de l'aplicació tenim que ha de ser una aplicació fàcil d'utilitzar, portable, flexible i ben organitzada. Per altre banda al ser una

aplicació web haurà de funcionar amb qualsevol navegador. La resolució que es farà servir serà 1280x1024; al ser una aplicació en la que es mostraran imatges la persona que visiti la web amb resolucions més baixes només notarà la diferencia en que caldrà veure menys imatges juntes a la pantalla.

2.4.3. Diagrames de casos d'us

A continuació es presenten els diagrames de la relació entre tot els casos d'us vistos en el punt anterior:

➤ **Part pública de la web**

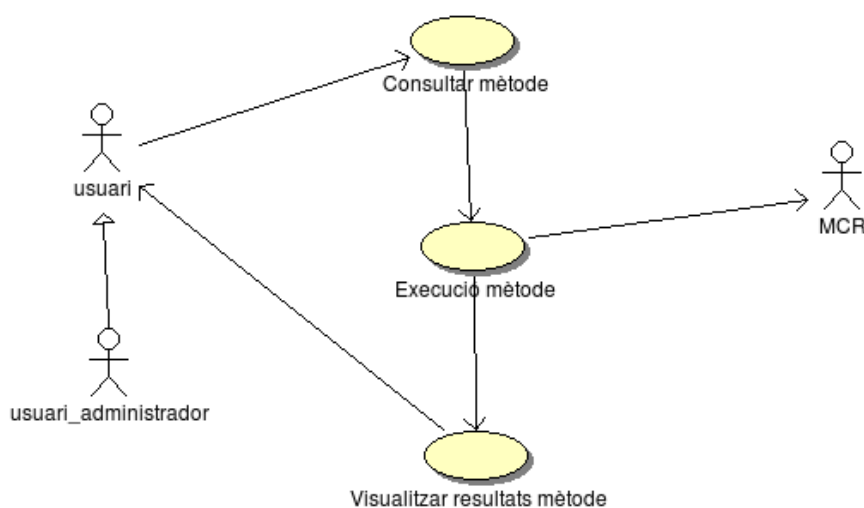


figura 4. Diagrama de casos d'ús part pública

➤ **Part privada de la web**

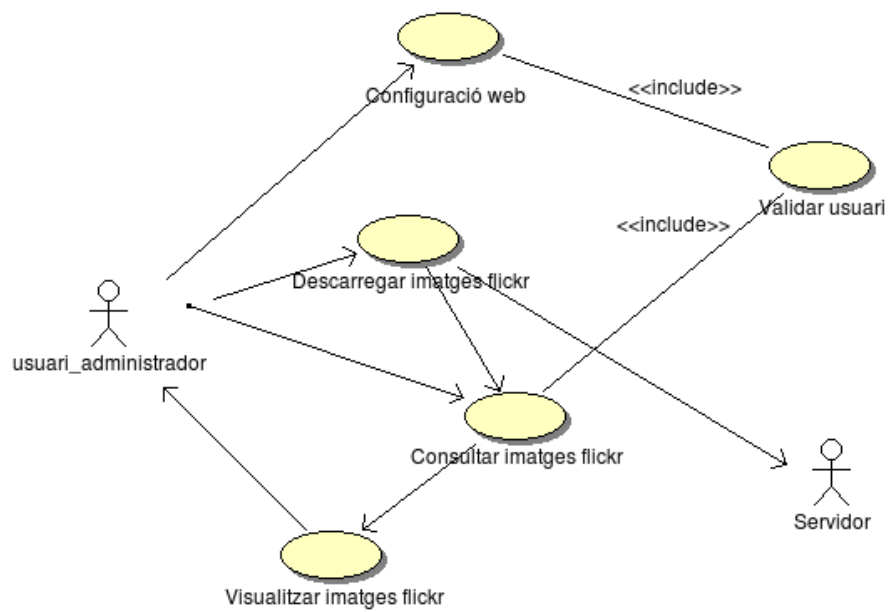


figura 5. Diagrama de casos d'ús part privada

2.4.4. Diagrames de seqüència

Aquests són tots els diagrames de seqüència de tots els escenaris de l'aplicació. Alguns dels escenaris ajunten diferents casos d'us:

- **Diagrama de seqüència del escenari consulta, execució i visualització d'un mètode (part pública):**

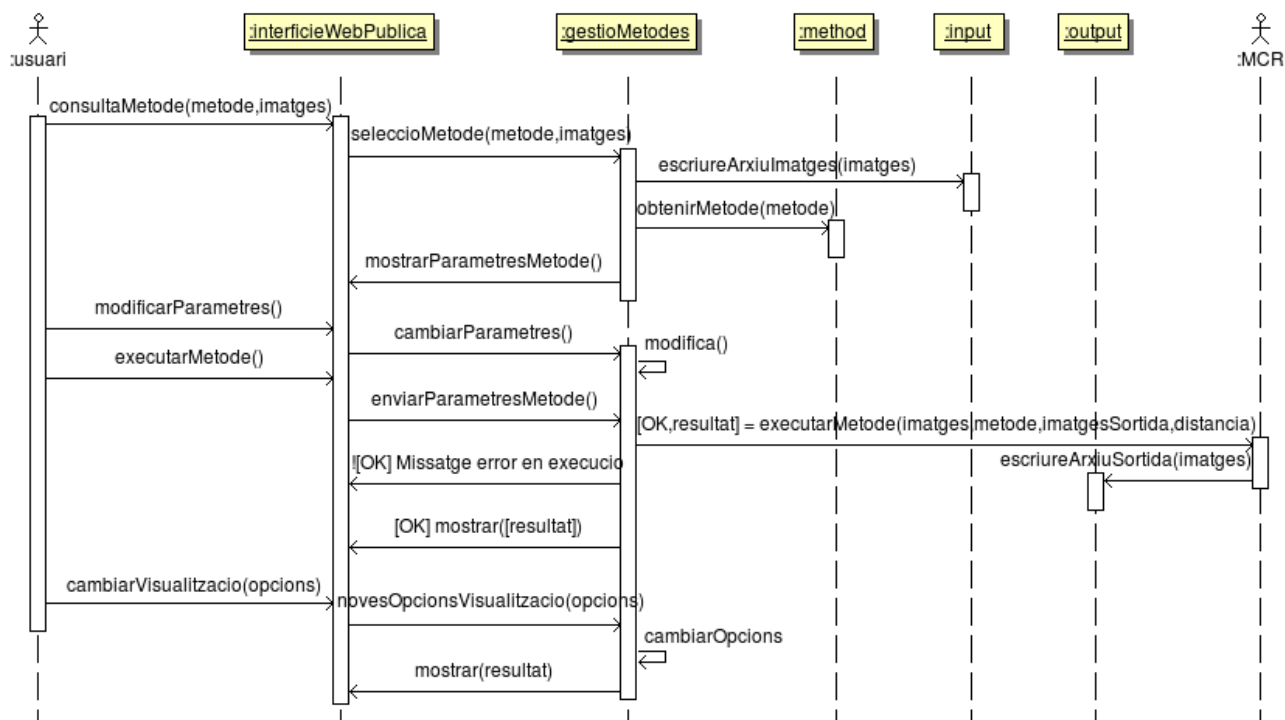


figura 6. Diagrama de seqüència consulta, execució i visualització d'un mètode

- **Diagrama de seqüència del escenari consulta, descàrrega i visualització d'imatges flickr (part privada):**

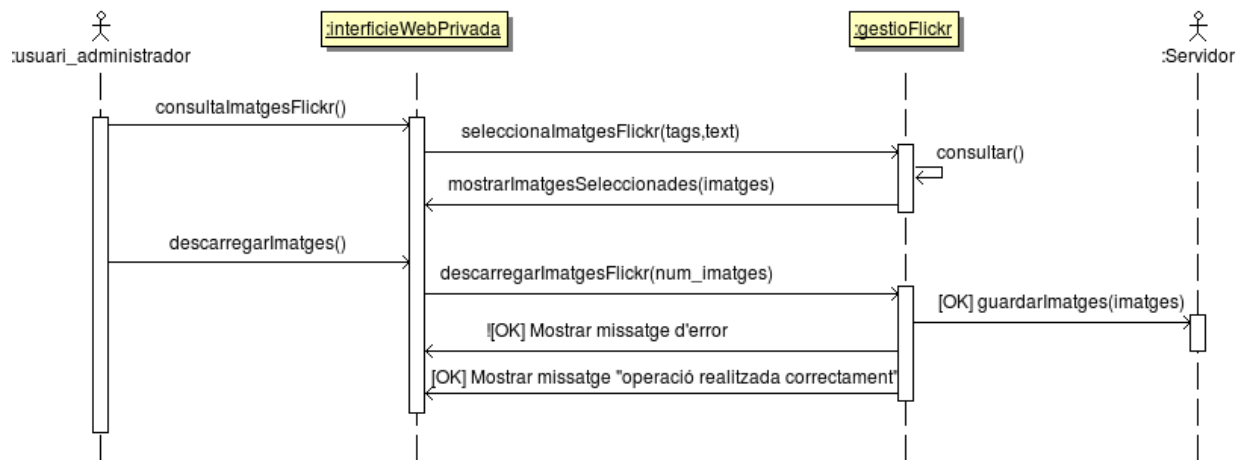


figura 7. Diagrama de seqüència consulta, descàrrega i visualització d'imatges flickr

- **Diagrama de seqüència del escenari configuració de la web (part privada):**

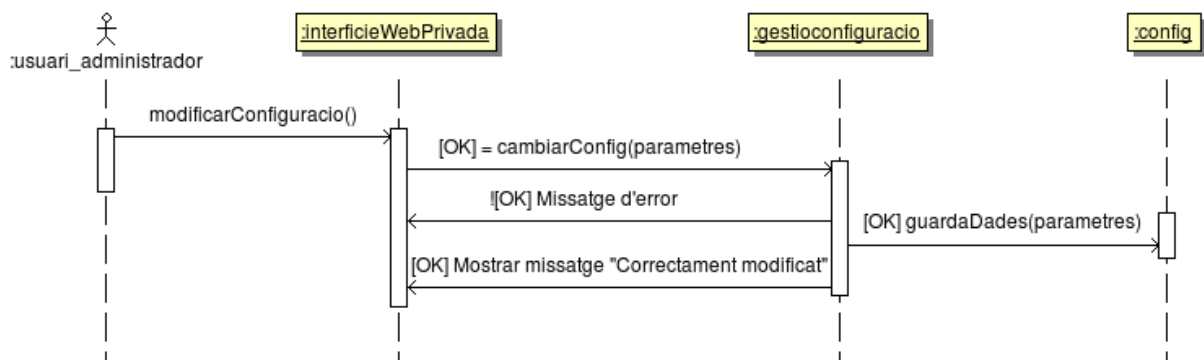


figura 8. Diagrama de seqüència configuració de la web

2.5. Eines de desenvolupament

Es farà una descripció de les eines que s'han utilitzat per fer el projecte. Per cada tipus d'eina es comentarà (si existeix) quines eren les possibilitats a triar i el perquè de la decisió. L'ordre en que es comentaran les eines és el mateix ordre en que es va anar decidint.

2.5.1. Sistema operatiu

Les opcions que es tenien eren 2 : Windows o Linux en les seves versions de servidor. Recordant el que es diu al punt de menú “motivacions” (1.2.) la filosofia d'aquest projecte és el software lliure per tan inicialment la decisió ja era clara, s'utilitzaria linux (Debian server 64 bits).



Figura 9. Logotips de sistemes operatius.

Per no complicar la compatibilitat entre sistemes es va decidir d'utilitzar un **Ubuntu server 32 bits**. D'aquesta manera el projecte també surt més rentable ja que no s'han de pagar les llicències, a diferència del que succeeix amb Windows.

2.5.2. Servidor web

El servidor web del grup ha sigut muntat en una màquina virtual dins de la màquina física que tal com s'ha dit en el punt anterior es tracta d'un servidor Debian. Per crear aquesta màquina virtual s'ha utilitzat el programa *vmware 2*, en el qual pots generar tantes màquines virtuals com vulguis, copiar altres màquines i visualitzar totes les màquines que hi han instal·lades (per configurar-les, parar-les o engegar-les) des d'una interfície web. En aquest cas només en tindrem una i serà el nostre servidor web.

En linux podem trobar entre d'altres dos dels més importants servidors web: apache i tomcat els dos són de , “*The apache software foundation*”. Són software lliure i existeixen tan per linux com per windows. Inicialment s'ha instal·lat un **servidor apache**.



Figura 10. Logotip apache foundation.

El servidor tomcat està més dedicat a servir pàgines programades en java en concret utilitzant les tecnologies *Java Servlet* i *JavaServer pages*. Com que inicialment en aquest projecte no es té previst fer les pàgines web en java es descarta aquesta opció.

2.5.3. Matlab a la web (cgi)

En aquest punt se'ns presenta una decisió important, escollir la manera en que executarem codi matlab des de web. Es presenten dues tecnologies alternatives.

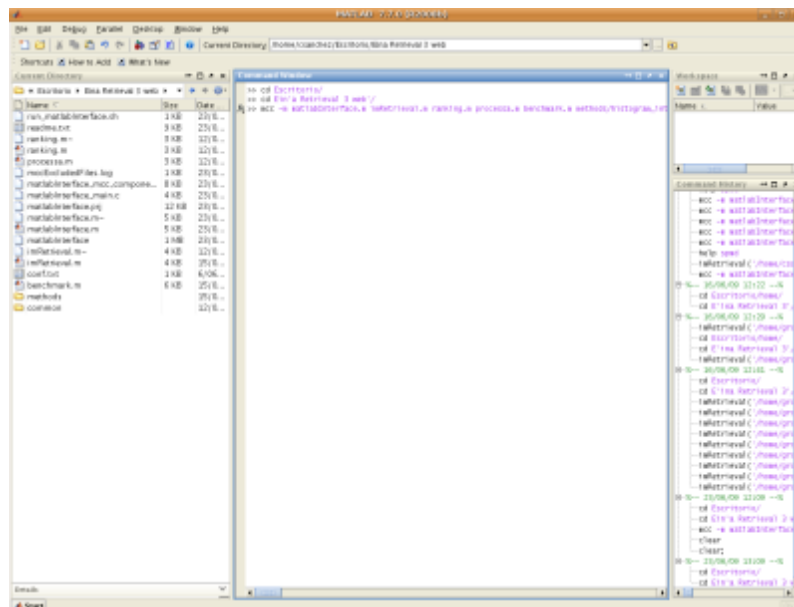


Figura 11. Interfície Matlab.

Per una banda el programa matlab incorpora varis paquets de funcionalitats anomenats *toolbox*. Una d'aquestes *toolbox* (Matlab Java builder JA) permet la creació de codi java per a web a partir d'un programa fet en matlab. És a dir, podem transformar el codi que tenim fet en matlab en un altre codi que utilitza la tecnologia comentada abans (punt 3.5.2 elecció del servidor) *Java Servlets*. Un cop tenim el nostre *Servlet* es diposita al servidor (tomcat) i s'executa com qualsevol altre web (a partir d'una direcció url). L'avantatge d'utilitzar aquesta tecnologia és que un cop tens ven programat el codi matlab per a web, funciona igual que com si l'estiguéssim executant des del propi programa, és a dir, es poden mostrar figures de gràfics , matrius , objectes ,... tot des del navegador. Per tan es podria posar qualsevol programa a la web.

Com a inconvenient tenim que la transformació del codi no és trivial. El procés de pas d'un llenguatge a un altre (matlab -> java) és automàtic però amb això només aconseguim que des de web es pugui executar el codi. El problema sorgeix quan es vol que es mostrin resultats (la majoria dels programes).

En aquest cas el que s'ha de fer es modificar el codi generat amb les comandes que siguin necessàries per a fer la mostra dels resultats.

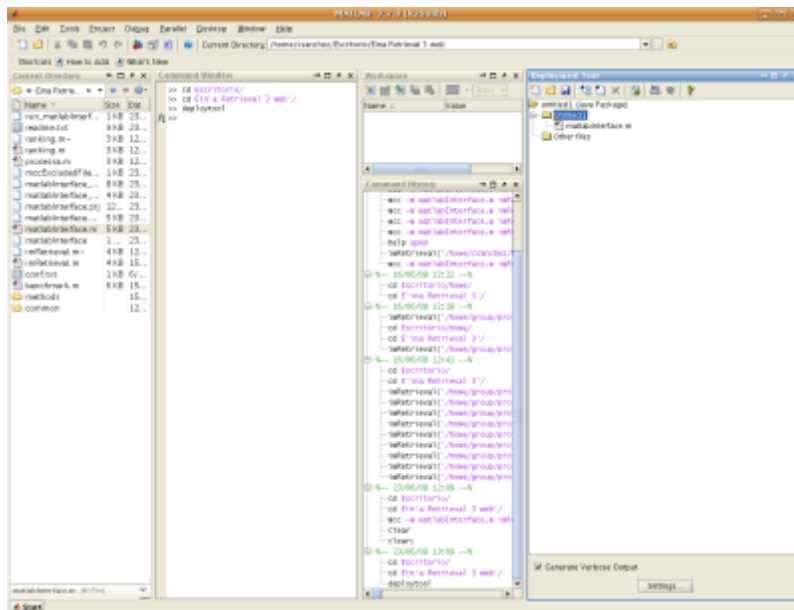


Figura 12. Matlab Builder JA columna de la dreta.

Per altra banda tenim l'opció d'utilitzar una funcionalitat dels servidor web, el *cgi* (*common gateway interface*), és una zona del servidor web on podem afegir executables i poden ser cridats via web.

Amb matlab podem compilar el codi d'un programa i crear el seu executable. Un clar avantatge, doncs, és la portabilitat del codi, ja que no ens em de preocupar a passar el codi que ja esta fet en matlab a un altre llenguatge. Com a inconvenient, però, tenim que al ser un executable totes les sortides del programa, a priori, no podran ser mostrades per web i per tan només podríem cridar l'execució del programa online.

La decisió doncs és difícil. Si ens agafem a dos dels objectius del projecte la solució és més clara.

Un dels objectius era la portabilitat de l'aplicació, en aquest aspecte la utilització del *cgi* seria la millor opció ja que no s'hauria d'obligar a ningú a aprendre java per a poder posar el seu programa a la web.

L'altre objectiu era la flexibilitat i el fàcil ús per a incorporar nou programari a la pàgina. Per a complir aquest objectiu la millor opció és *cgi* ja que la persona que vol incorporar un nou programa només l'hauria de compilar per genera el seu executable i

afegir-lo al directori cgi del servidor.

La funcionalitat que **s'utilitzarà** per a executar programes matlab des de web serà **cgi**. S'ha de tenir en compte que la millor opció per mostrar els resultats d'un programa a la web és utilitzant la toolbox del matlab, per tan s'haurà de buscar una metodologia de treball eficient per aconseguir resultats semblants fent servir executables de cgi. El problema doncs el tindrem en com aconseguir que les sortides d'un executable puguin ser mostrades a la web. Això s'explicarà en la secció de disseny de la memòria.

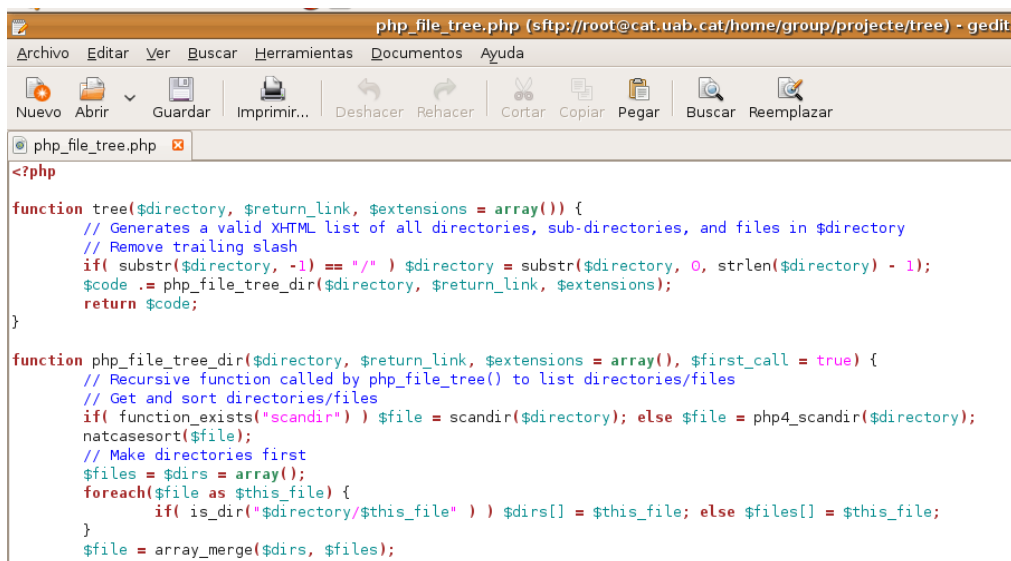
2.5.4. Llenguatges de programació

El projecte es basa en una aplicació web, a més ja s'ha comentat la necessitat d'utilització de software lliure, per tan la tria dels llenguatges de programació està molt acotada. Per a la programació de les pàgines web s'utilitzarà principalment **html** acompanyat de **php** (per a la generació i programació de pàgines dinàmiques) i **javascript** per a les funcionalitats executades al ordinador client (missatges d'error, visualitzacions d'estil , etc). Per al disseny de l'estil utilitzarem **css** (Cascading Style Sheets) que ens permet poder canviar (en el moment que es vulgui) el format de totes les pàgines modificant tan sols un arxiu. Aquesta és una combinació de llenguatges molt típica en la programació de sites.

Per a la configuració de la web s'utilitzarà un arxiu en **xml**.

Per a la programació d'algun procés al servidor (explicat a la secció de disseny) s'utilitzarà el llenguatge de programació anomenat **perl** , molt usual en scripts per al manteniment de servidors.

Per últim també es farà servir **matlab** per crear les funcions que ens siguin necessàries per a la comunicació entre web i els programes fets en aquest llenguatge. Tots aquests llenguatges han sigut programats directament amb el **gedit** de linux. Un editor de textos amb sintaxi per molts llenguatges entre ells tots els anomenats anteriorment.



The screenshot shows the gedit text editor with a file named 'php_file_tree.php'. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Buscar', 'Herramientas', 'Documentos', and 'Ayuda'. The toolbar contains icons for 'Nuevo', 'Abrir', 'Guardar', 'Imprimir...', 'Deshacer', 'Rehacer', 'Cortar', 'Copiar', 'Pegar', 'Buscar', and 'Reemplazar'. The code in the editor is as follows:

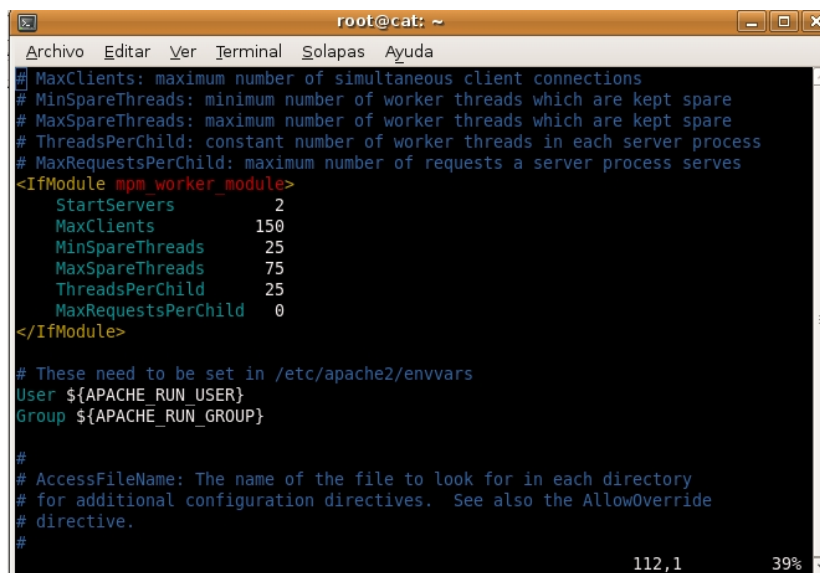
```
<?php

function tree($directory, $return_link, $extensions = array()) {
    // Generates a valid XHTML list of all directories, sub-directories, and files in $directory
    // Remove trailing slash
    if( substr($directory, -1) == "/" ) $directory = substr($directory, 0, strlen($directory) - 1);
    $code .= php_file_tree_dir($directory, $return_link, $extensions);
    return $code;
}

function php_file_tree_dir($directory, $return_link, $extensions = array(), $first_call = true) {
    // Recursive function called by php_file_tree() to list directories/files
    // Get and sort directories/files
    if( function_exists("scandir") ) $file = scandir($directory); else $file = php4_scandir($directory);
    natcasesort($file);
    // Make directories first
    $files = $dirs = array();
    foreach($file as $this_file) {
        if( is_dir("$directory/$this_file" ) ) $dirs[] = $this_file; else $files[] = $this_file;
    }
    $file = array_merge($dirs, $files);
```

Figura 13. Programant php amb gedit.

Per a la configuració del servidor web s'ha utilitzat un editor per consola de linux, anomenat VIM. Aquest editor també té diferents sintaxis per a diferents llenguatges de programació.



The screenshot shows the VIM text editor in a terminal window. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Terminal', 'Solapas', and 'Ayuda'. The code in the editor is as follows:

```
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_worker_module>
    StartServers      2
    MaxClients       150
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadsPerChild   25
    MaxRequestsPerChild 0
</IfModule>

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives. See also the AllowOverride
# directive.
#
```

figura 14. Configurant el apache des de VIM.

2.5.5. Bases de dades

A l'aplicació hi han moltes dades que han de ser guardades permanentment, com per exemple la configuració del site o els components que formaran part del formulari de consulta d'un mètode en concret. Per aquesta raó es necessita una base de dades on guardar les dades que vulguem mantenir. Novament tenim dues opcions : fer servir el sistema gestor de base de dades mysql o guardar totes les dades en arxius. Inicialment es va començar utilitzant mysql però en el moment en que es vol per exemple muntar la web en un altre servidor s'ha de tenir en compte d'instal·lar el mysql i d'exportar i importar la base de dades i això només ho podria fer un expert. També està el problema de que donat el cas ens volem emportar els mètodes creats per executar-los des de una altra pàgina web que utilitzés el mateix sistema també es necessitaria alguna persona que ho gestionés.

Per altra banda tenir una base de dades en mysql provocaria haver de crear pantalles addicionals de manteniment de les taules i això no es d'interès per als membres del grup.

Tenint en compte el que s'ha dit, l'opció d'utilitzar mysql es va desestimar i per tan el que s'ha fet servir per a **base de dades són un conjunt d'arxius**. La portabilitat i fàcil configuració és clara, l'únic que s'hauria de fer per a traslladar configuracions o mètodes a altres aplicacions seria copiant aquests arxius i dipositant-los allà on tocava , sense necessitar l'assistència com ens passava amb la base de dades mysql.

La configuració de tots els arxius encarregats de ser la base de dades està definida en la secció de 'disseny' de la memòria.

3. DESENVOLUPAMENT

3.1. Metodologia de treball

L'elaboració d'aquest projecte es farà de forma gradual, per tant podem dir que seguirà un model en espiral incremental de desenvolupament.

La primera etapa on recopilarem tota la informació necessària de les necessitats del grup de recerca a l'hora de presentar els seus programes. És a dir, veure quines característiques comunes tenen els programes que fan, per crear un model que ens serveixi per la majoria de mètodes o programes que es vulguin posar a la web.

Una segona etapa de recerca per saber com utilitzar totes les eines que s'ha decidit fer servir.

L'última etapa és el disseny de totes les parts del projecte assegurant-nos que cada part funciona correctament , i finalment la unió de totes elles per crear un prototip inicial que, depenent del temps , s'anirà treballant sobre ell per anar-lo millorant. En aquest punt s'ha tornat a definir algunes especificacions i/o objectius inicials ja que algunes de les especificacions que s'han fet al principi, han sigut adaptades un cop s'anaven aconseguint nous prototips i s'anaven descobrint noves necessitats de l'aplicació.

3.2. Fonaments teòrics

A part de tot el programari que es farà servir per desenvolupar aquesta aplicació (comentat al punt 3.5 eines de desenvolupament) es necessiten unes bases teòriques per a poder començar a desenvolupar l'aplicació. Aquestes bases no és res més que la familiarització amb diferents punts en els quals no es tenia experiència prèvia. Aquests coneixements a assolir són el que es comenta tot seguit:

3.2.1. Programari matlab via web

Per poder arribar a executar programes matlab des de web, com s'ha aconseguit, s'ha tingut que fer una cerca i aprenentatge de les tècniques necessàries. Al punt anteriorment descrit (3.5 Eines de desenvolupament) ja s'ha decidit que la eina a utilitzar serà el cgi.

El cgi (cammon getaway interface) és una important tecnologia del *World Wide Web* (www) que permet que un client (navegador web) sol·liciti dades d'un programa executat al servidor. CGI especifica un estàndar per a transferir dades entre el client i el programa. És un mecanisme entre el servidor web i una aplicació externa. Les aplicacions que s'executen reben el nom CGIs.

Per crear el nostre CGIs dels nostres mètodes en matlab es va consultar l'ajuda d'aquest programa (Matlab) en d'altres fonts d'informació. La manera d'aconseguir afegir un codi matlab com a cgi al servidor és compilant els mètodes per aconseguir executables. Aquests executables poden ser tractats com a cgi ja que són una aplicació. En matlab la compilació d'un mètode s'aconsegueix amb la comanda:

```
mcc -m nomMetodeAcompilar.m
```

El directori definit al servidor per afegir totes les aplicacions (CGIs) es troba:

```
/usr/local/cgi-bin/
```

En aquest directori podem afegir tants subdirectoris com vulguem i tots ells seran tractats com a CGIs.

Amb això ven après l'únic que ens falta entendre i veure el funcionament, és com fer la crida d'aquests CGIs des d'una pàgina web o el que és el mateix des d'un navegador web. El sistema no és gens complicat i menys per a persones ja familiaritzades amb el llenguatge html. Consisteix en crear un formulari (html) amb tots els camps que necessiti el CGIs que ja tenim al directori corresponent. El formulari farà la crida aquest CGIs i no a un altre pàgina com normalment es fa.

Aquest es un exemple de formulari en html que crida a un CGI's concretament un programa que ja ha sigut compilat en matlab:

```
<form id="rad" method="get" action="/cgi-bin/template/webTemplate" name="frm2" >
  <input type="hidden" name="name" SIZE="50" >
  <p>thr: <input name="thr" type="text" value="2" size="5"></p>
  <input type="SUBMIT" VALUE="Go!">
</form>
```

figura 15. Crida d'un CGI's des de web.

Com es pot observar el mètode d'enviament de dades és *get* i és així per enviar el valors dels camps del formulari per la url. Ha quina url encadena aquestes dades que envia? doncs a la direcció que apareix a *action*, que com es pot veure en el nostre cas, és una aplicació que tenim com a CGI's i que es troba en el directori comentat abans.

L'últim a aprendre i per finalitzar aquest apartat, ha sigut com fer que un mètode matlab compilat, pugui llegir les dades que un formulari web li està enviant.

Les dades arriben a l'aplicació CGI's per mitjà d'una variable d'entorn: un conjunt de valors dinàmics que normalment afecten al comportament dels processos en un ordinador. Aquesta variable concretament es diu *QUERY_STRING* i el seu valor s'aconsegueix utilitzant la funció *getenv()*, que és per recuperar variables d'entorn. Amb això ja tenim les dades enviades des del formulari.

Al capítol de disseny s'explicarà quin i com és el tractament que se li fan aquestes dades i com es retornen les respostes a l'aplicació.

3.2.2. Api de flickr

Inicialment no es coneixia de l'existència d'algun mètode per a consultar la base de dades de flickr, i puguer utilitzar les seves imatges. La primera notícia va arribar a partir de la lectura d'una publicació:

James Hays, Alexei A. Efros. IM2GPS: estimating geographic information from a single image. Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

El programa del qual parla la publicació consisteix en rebre una imatge com a

paràmetre i trobar quina és la situació d'aquesta imatge en un mapa mundial. Això ho aconseguirien buscant a flickr les imatges més semblants a la que s'està passant per paràmetre i descobrir així, en quin lloc del món van ser preses. El que ens importa de tot això no és el que fa el programa sinó com s'ho fan per a consultar les imatges de flickr. Mirant la documentació de l'aplicació es va trobar que el propi flickr dona una API (en diferents llenguatges) per a poder fer la comunicació des de un programa extern, a la seva base de dades.

A trets generals per poder realitzar la comunicació és necessari tenir una clau ID per a poder utilitzar la API, aquesta, es pot aconseguir donant-se d'alta a flickr. La utilització d'aquesta API, es permet sempre i quan el us no sigui comercial.

Tota la documentació referent a quines són les funcionalitats i com són les sintaxis de totes les seves funcions, es pot trobar a : <http://www.flickr.com/services/api/>

Com es pot veure flickr ofereix la API en diferents llenguatges, perquè així, es puguin fer les consultes des de diferents programes externs. En el nostre cas, i tenint en compte que es tracta d'una aplicació web, utilitzarem la API en php.

3.2.3. Característiques dels mètodes matlab

El que es vol és crear una aplicació el màxim de flexible. Com a flexible ens referim a que pugui ser ampliada (en número de mètodes) sense que això impliqui canvis en la implementació de l'aplicació web. Per aquesta raó ha sigut necessari resumir un conjunt de característiques que seran globals per a tots els mètodes. Això s'ha aconseguit a partir de reunions amb els membres del grup, els quals han exposat les seves necessitats.

Finalment s'ha arribat a una llista resum de les característiques comunes que ha de tenir un mètode per poder ser afegit a la web:

- Els **mètodes sempre consulten sobre un conjunt d'imatges**.
- Els **mètodes sempre necessiten fer un conjunt de càlculs previs** sobre cada imatge d'aquest conjunt.
- Poden tenir un **número invariant de paràmetres**.
- **Un dels paràmetres sempre serà una imatge** (que podrà ser definida o no) amb la qual es vol fer la comparació amb tot el conjunt d'imatges amb el que es

consulta.

- **Un altre dels paràmetres sempre serà el nom d'una distància**, que és l'equació que es farà servir per calcular quines són les imatges més properes després d'aplicar el mètode.
- La **sortida dependrà de si s'ha definit la imatge d'entrada**:
 - **Si existeix** la imatge com a paràmetre, la **sortida** del mètode **serà el conjunt d'imatges** amb el que es volia consultar **però amb un cert ordre** i amb un conjunt de puntuacions associades a cada imatge.
 - **No existeix** la imatge com a paràmetre la **sortida** del mètode **serà un gràfic i una descripció resum** del que ha sigut el anàlisi del conjunt d'imatges amb el mètode escollit.

Per afegir un mètode més flexiblement, s'haurà de pensar una interfície en la qual es cridi al mètode seleccionat a la web, amb independència del número de paràmetres que necessita. L'únic que és obligatori complir són els mínims explicats en aquest punt.

3.2.4. Programació

Per a la programació de la web i tots els seus components s'han utilitzat el coneixements previs que ja es tenien en alguns llenguatges com html, php i javascript. En aquests llenguatges s'han tingut que aprendre noves tècniques que es desconeixien com per exemple creació de sessions, utilització de *cookies* per al pas d'informació, creació/modificació d'objectes des de javascript, crides a aplicacions cgi, o la lectura d'arxius amb una certa nomenclatura. Tots els detalls de disseny estaran comentats a la secció corresponent.

S'han utilitzat altres llenguatges, els quals no s'havien utilitzat mai:

- **Perl** , llenguatge que s'utilitzarà per a crear scripts interns per al servidor. L'utilitzarem per a crear un script propi per a la càrrega d'imatges al servidor. Tota la informació sobre la sintaxi s'ha extret de fòrums i pàgines de programació.
- **Matlab** , aquest llenguatge, serà utilitzat per a crear una interfície en matlab per a poder fer la crida de tots els mètodes que siguin integrats a la web. Aquesta crida haurà de ser el mes flexible possible, ja que, si es volen afegir

nous mètodes, això no ens provoqui que tinguem de modificar aquest codi. Tota l'ajuda necessària per a fer servir aquest llenguatge s'extraurà de l'ajuda del propi programa i de la seva pàgina web (<http://www.mathworks.com/>).

- **Css** , llenguatge de programació que serà utilitzat per aplicar un estil a la pàgina web, tot centralitzat en un únic arxiu.

El funcionament de totes les tasques programades en cada llenguatge serà definit a la secció de disseny de la memòria.

Des del principi d'aquesta memòria que s'està parlant que serà un projecte per al grup de recerca CAT. La llengua oficial en aquest grup, com en la majoria dels grups científics, és l'anglès. Així doncs el codi d'aquest projecte, amb independència del llenguatge utilitzat, serà comentat en anglès.

3.3. Anàlisi

3.3.1. Organització de la web

Des de el punt de vista de l'usuari de la web, aquesta és divideix en dos blocs:

- **Part pública**, en aquesta part els usuaris poden provar mètodes i veure els resultats provant amb diferents paràmetres.
- **Part privada**, aquesta part està protegida per a que només els usuaris administradors (donats d'alta al servidor) puguin configurar la web. També en aquesta part els usuaris poden crear-se les seves pròpies bases de dades d'imatges descarregant-les des de flickr.

A continuació es comenten quines tècniques s'utilitzaran per aconseguir enviar tota la informació necessària des de la web als programes en matlab i la lectura dels resultats que ens donen les execucions d'aquest programes:

- **Conjunt d'imatges**, com ja s'ha comentat tots els mètodes esperen un conjunt d'imatges per a treballar sobre elles. Aquestes imatges seran enviades als programes matlab a partir de la direcció d'un arxiu que contindrà les adreces de

totes les imatges amb les que s'han de treballar. Per tan el paràmetre que farà referència a les imatges serà l'adreça d'un arxiu.

- **Conjunt imatges resultant**, també es passarà com a paràmetre l'adreça de l'arxiu on el programa en matlab escriurà les adreces de totes les imatges un altre cop. Aquesta vegada, però, amb l'ordre en que el programa executat decideix que han de ser mostrades. Després de l'execució del programa, la web ja sabrà quin és l'arxiu on està el resultat del mètode.
- **Paràmetres del mètode**, estaran definits en arxius únics per a cada mètode i seran representats com a camps del formulari.
- **Imatge de consulta**, serà un camp més del formulari del mètode i podrà estar compostat entre 0 i n imatges.

A continuació es presenten el ànàlisi de quina hauria de ser la divisió de la web per arxius i quina funció hauria de tenir cada un d'ells:

- **index** Menú inicial de la web en el qual pots consultar sobre un mètode i un conjunt d'imatges (carrega la pàgina `methodData`). També hi haurà l'opció d'anar a la part privada de la web.
- **methodData** pàgina en la que es generaran els formularis dels mètodes. Des de aquesta pàgina s'enviaran les dades al MCR per executar el mètode seleccionat.
- **ImageReziFunctions** paquet de funcions per generar thumbnails (és utilitzat des de la pàgina en que es mostren els resultats).
- **answer** pàgina que generarà la sortida dels mètodes executats (els resultats).
- **redirecciona** pàgina que serà cridada des de el codi Matlab. Des de aquesta pàgina cridarem al `answer`.
- **calculating** pàgina que es cridarà des de l'inici dels càlculs dels mètodes per indicar que el procés pot trigar una estona.
- **config** pàgina que mostra el formulari de modificació de paràmetres de l'aplicació.
- **readXMLconf** paquet amb la funció d'extreure els valors del arxiu de configuració de la web.
- **writeXMLconf** paquet amb la funció que guarda els nous valors en el arxiu de configuració de la web.
- **index** (de consultes a flickr) pàgina en la que a partir d'un formulari de filtratge

executem les consultes d'imatges a la base de dades de flickr. Des de aquesta pàgina es crida a la pàgina search.

- **search** pàgina que mostra totes les imatges de la consulta realitzada a flickr.
- **download** paquet amb les funcions de descàrrega d'imatges de flickr al servidor.

3.3.2. Funció dels components de la web

En aquesta secció es presentaran quins són tots els components que interaccionen amb la web. Aquests es poden dividir en dos grans grups:

- **Components que formen part de formularis de mètodes** per a la creació dels formularis amb tots els paràmetres de cada mètode. Per a la representació d'aquests paràmetres s'utilitzaran els següents components:
 - **text** component propi dels formularis en html. Mostra els paràmetres que són un text ([A..Z][0..9][.,]).
 - **combobox** llista de valors. És un component propi dels formularis en html. S'utilitzarà en aquells paràmetres amb varies opcions concretes.
 - **checkbox** opció entre cert o fals. Component propi dels formularis en html. S'utilitzarà en aquells paràmetres en que el valor pot ser 'cert' o 'false'.
 - **upload** component creat per a la pujada d'imatges al servidor. S'utilitzarà en el paràmetre que ha d'existir en tots els mètodes, la imatge de consulta, que pot estar definida o no.
 - **colourPicker** component creat per a la selecció de colors a partir d'una paleta. S'utilitzarà en aquells paràmetres en que es necessiti el codi d'un o varis colors.
 - **tree** component creat per a la selecció de directoris de conjunts d'imatges, per a ser analitzades, amb els mètodes. Aquest component genera un arbre de directoris.
- **Components de suport a l'aplicació** , tals com, arxius de configuració , bases de dades o components de comunicació.
 - **matlabInterface** component encarregat de rebre peticions de consultes de mètodes, executar-los, i retornar els resultats.
 - **uploadMulti** component instal·lat al servidor encarregat de pujar les

imatges de consulta al servidor. Aquest component es crida per component de formulari, upload.

- **phpFlickr** component amb el paquet de funcions de l'API de flickr. Encarregat de fer les consultes a la base de dades d'imatges de flickr i retornar els resultats.
- **config.xml** arxiu de configuració de la web. Guarda tota la configuració de la web.
- **arxius .method arxius** de configuració dels mètodes. Seran llegits per a construir els formularis pertinents.
- **arxiu distance.dis** arxiu on estan definits els noms de les possibles distàncies a utilitzar. S'entén per a distància, la funció de càlcul de la diferència entre característiques d'imatges.
- **arxius input** arxius que es generaran amb les adreces del conjunt d'imatges seleccionades per a ser analitzades.
- **arxius output** arxius que generaran l'execució dels mètodes amb les adreces de les imatges en ordre com han de ser mostrades. En el cas que no hi hagi una imatge de consulta aquests arxius contendran el text a mostrar per la web.

3.3.3. Diagrama web-components

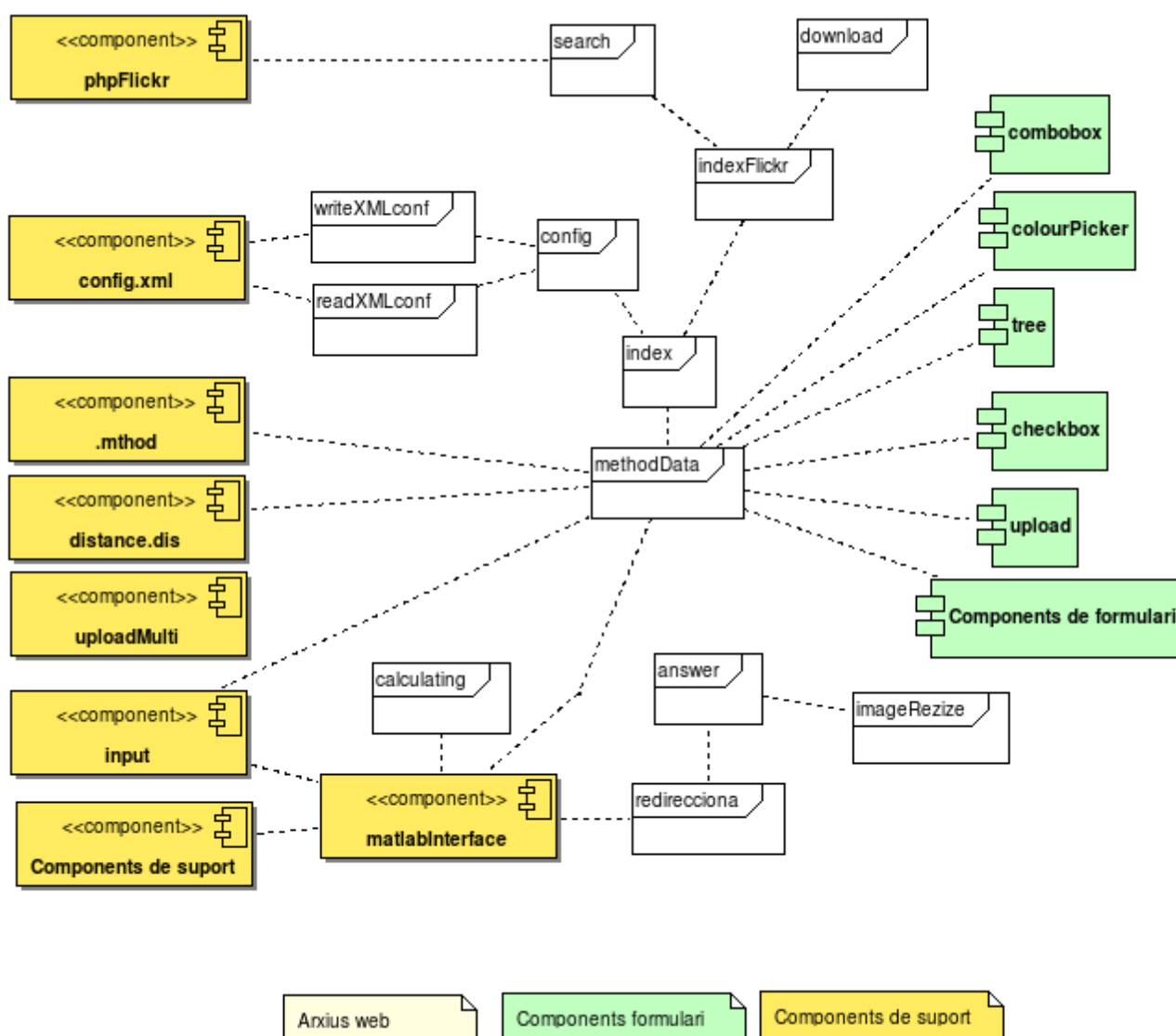


figura 16. Diagrama de comunicacions de la web.

En l'anterior diagrama podem observar totes les relacions de comunicació entre tots els arxius que componen la web i els components que intervenen. Aquestes connexions estan representades per línies discontinues. Per seguir correctament el diagrama s'ha de tenir present que l'arxiu inicial de la web és el index i és a partir d'aquest d'on s'inicien tots els fluxs. Totes les unions de comunicació es consideren com a possibles crides a un component o arxiu de la web.

4. DISSENY I IMPLEMENTACIÓ

4.1. Fases

En aquest apartat s'explica quin a sigut el procés de construcció de la web cronològicament ordenat. Per a cada fase es comenta i es mostra quines són les configuracions, implementacions i visualitzacions corresponents.

4.1.1. Muntatge del servidor web

En aquesta secció es farà un resum de les configuracions, paràmetres o mòduls que s'han tocat per aconseguir el funcionament del servidor web adaptat a les necessitats que es tenen. Aquestes són les característiques:

- **Tipus de servidor web** : s'ha instal·lat un **apache2** per a fer la funció de servidor web. L'únic que s'ha canviat, a la seva configuració per defecte, ha sigut, el *timeout*. Aquest és el temps d'espera del servidor a una resposta. Aquest temps s'ha augmentat a 30 minuts per als processos d'execució de mètodes.
- **Port** : 80.
- **Mòduls instal·lats** : per al correcte funcionament de la web s'han instal·lat 2 mòduls:
 - **php5** , aquest mòdul ha sigut instal·lat per a que el servidor pugui llegir pàgines programades en php, i generar el codi html que mostrarà el navegador. En el arxiu de configuració (`php.ini`) només s'ha canviat una opció: `max_upload_file = 20 Mb` , per així no tindre problemes amb els tamanyes de les imatges que es pujaran al servidor.
 - **Gd2** , aquest mòdul s'ha instal·lat i activat per al tractament d'imatges. En concret l'aplicació l'utilitzarà per la creació de thumbnails, procés en el qual s'ha de redimensionar la imatge però sense perdre informació.
- **Variables d'entorn** : el programes en matlab compilats necessiten saber les variables d'entorn tal i com si estiguessin executant-se en el entorn matlab. Per poder executar els programes des de web es necessari que aquestes variables

siguin definides en el servidor. La següent figura mostra la configuració del arxiu *httpd.conf* del servidor amb les corresponents variables d'entorn definides:

```
setEnv MCRROOT /opt/MATLAB/MCR/v79
setEnv MCRJRE /opt/MATLAB/MCR/v79/sys/java/jre/glnx86/jre/lib/i386
setEnv LD_LIBRARY_PATH ./opt/MATLAB/MCR/v79/bin/glnx86:/opt/MATLAB/MCR/v79/runtime/glnx86:/opt/MATLAB/MCR/v79/sys/os/glnx86:/opt/MATLAB/MCR/v79/sys/java/jre/glnx86/jre/lib/i386/native_threads:/opt/MATLAB/MCR/v79/sys/java/jre/glnx86/jre/lib/i386/server:/opt/MATLAB/MCR/v79/sys/java/jre/glnx86/jre/lib/i386/client:/opt/MATLAB/MCR/v79/sys/java/jre/glnx86/jre/lib/i386
setEnv XAPPLRESDIR $MCRROOT/X11/app-defaults
```

figura 17. Configuració del *httpd.conf* del apache.

- **Definició del directori base de la web** : la configuració del directori base de la web s'ha realitzat en el arxiu */etc/apache2/sites-enabled/000-default* i és la següent:

```
ServerAdmin webmaster@localhost

DocumentRoot /home/group/projecte
<Directory />
    Options FollowSymLinks MultiViews IncludesNoExec
    AllowOverride FileInfo
</Directory>
```

figura 18. Configuració del *000-default* del apache.

- **Protecció part privada** : el accés a la part privada de la web ha d'estar restringit a només usuaris que estiguin donats d'alta en el servidor. Per això s'ha fet una configuració addicional en la qual es protegeix el directori *private* del servidor i que és la intranet de la web. Aquesta configuració provoca que quan un usuari intenta entrar algun arxiu sota el directori protegit, necessitarà posar la contrasenya que tingui en el servidor. La configuració és la següent:

```
<Directory /home/group/projecte/private>
    AuthName "Restricted site"
    AuthType Basic
    AuthUserFile /dev/null
    AuthBasicAuthoritative Off

    AuthShadow on
    require group color
</Directory>
```

figura 19. Configuració de restriccions del apache.

4.1.2. Configuració dels components

En aquest capítol es tornarà a fer referència tots els components comentats en el capítol 3.3.2 però en aquest cas l'explicació es centrarà en quina és la forma d'utilització, com es defineixen i com es visualitzen.

- **Components que formen part de formularis de mètodes** els 5 primers components són tots els possibles per definir mètodes. És a dir, son totes les opcions disponibles per a representar mètodes en formularis web. Aquests formularis es defineixen en un arxiu .method que en següents seccions s'explicarà el seu funcionament.

- **text** component propi dels formularis html.

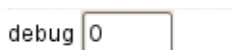
Sintaxi:

TEXT;nom_paràmetre;descripció_paràmetre;valor;tamany

Exemple:

TEXT;debug;debug;0;3

visualització:



- **combobox** component propi dels formularis html. Pot ser utilitzat per a paràmetres que tinguin un nombre fix de possibilitats.

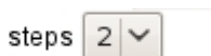
Sintaxi:

COMBOBOX;nom_paràmetre;descripció_paràmetre;número_de_valors
;valor_per_defecte ;altres_valors;...

Exemple:

COMBOBOX;steps;steps;2;2;1

visualització:



- **checkbox** component propi dels formularis html. Pot ser utilitzat per a paràmetres que tinguin com a valor cert o fals.

Sintaxi:

CHECKBOX;nom_paràmetre;descripció_paràmetre;valor(true,false)

Exemple:

CHECKBOX;synergistic;synergistic>true

visualització:

synergistic true ☒ false ☐

- **upload** component creat específicament per a la funció de càrrega d'imatges al servidor. Aquest component permet pujar entre 0 i n imatges. Per la posterior utilització en l'execució dels mètodes.

Sintaxi:

UPLOAD;nom_del_paràmetre

Exemple:

UPLOAD;imatge

visualització:



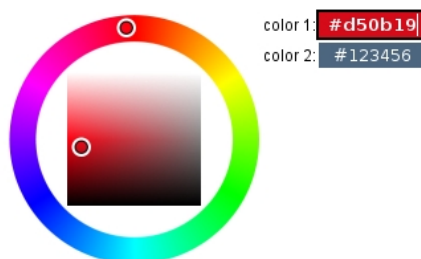
- **colourPicker** component creat per a la selecció de colors a partir d'una paleta. En alguns mètodes es necessita fer consultes sobre les bases de dades d'imatges filtrant per algun o varis colors. Aquest component ha sigut creat per poder passar com a paràmetre als mètodes, tants colors com vulgui.

Sintaxi:

PALETA;num_colors;nom_color1;desc_color1;
nom_color2;desc_color2;...

Exemple:

PALETA;2;color1;color 1;color2;color 2

visualització:

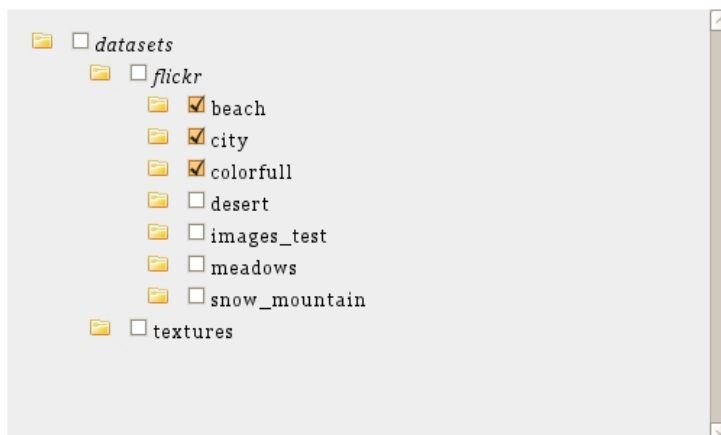
- **tree** component creat per a la selecció de directoris de conjunts d'imatges. La funcionalitat d'aquest component és, que donat un directori es genera un arbre d'exploració de tots els directoris que conté en el seu interior. Cada un d'aquests directoris porta associat un checkbox. D'aquesta manera els usuaris podran triar amb quines bases de dades d'imatges volen executar el mètode.

Com s'inicialitza:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="modules/tree/styles/default/default.css" rel="stylesheet" type="text/css" media="screen" />
<link href="http://www.cat.uab.cat/projecte/style/style.css" rel="stylesheet" type="text/css" />
<!-- Makes the file tree(s) expand/collapse dynamically -->
<script src="modules/tree/php_file_tree.js" type="text/javascript"></script>
```

Com es crida:

```
echo tree($_SERVER['DOCUMENT_ROOT'].$dirProject."data/images", "javascript:alert('You clicked on [link]');", $ext);
```

visualització:

➤ **Components de suport a l'aplicació**

- **arxius .method** són els arxius en els quals es configuren els mètodes. Aquests arxius estan compostat a partir dels components de formulari vistos en el punt anterior. Són llegits des de la web per a construir dinàmicament els formularis que executaran el mètode representat.

Exemple de configuració:

```
11
UPLOAD;imatge
TEXT;featurefun;featurefun;@RGB2Luv;10
COMBOBOX;steps;steps;2;2;1
CHECKBOX;synergistic;synergistic;true
TEXT;SpatialBandwidth;SpatialBandwidth;7;3
TEXT;RangeBandwidth;RangeBandwidth;6.5;3
TEXT;MinimumRegionArea;MinimumRegionArea;20;3
COMBOBOX;SpeedUp;SpeedUp;3;2;1;3
TEXT;GradientWindowRadius;GradientWindowRadius;2;3
TEXT;MixtureParameter;MixtureParameter;0.3;3
TEXT;EdgeStrengthThreshold;EdgeStrengthThreshold;0.3;3
```

Com es pot veure la primera línia defineix el número de paràmetres que contindrà el formulari representatiu del mètode. Totes les següents línies representen cada un dels camps.

- **arxiu distance.dis** és un sol arxiu del qual s'extreuen el nom de les funcions de distància que s'utilitzaran. És un paràmetre obligatori que sempre està present en tots els formularis. Cada una d'aquestes distàncies

és una forma de diferent de calcular semblances entre característiques de les imatges processades pels mètodes. Aquest càlcul de la distància forma part del propi mètode. Per afegir una nova distància (sempre i quan el mètode estigui preparat per utilitzar-la) només cal afegir el nom en aquest arxiu.

Definició de l'arxiu:

```
euclidean
mahalanobis
bhattacharyya
histogram_intersection|
```

Representació en el formulari: es carreguen tots els noms en un quadre de text.

Distance

- **arxiu input** aquest arxiu es genera en el moment de seleccionar quin és el conjunt d'imatges amb els que es vol provar el mètode. S'examinen tots els directoris seleccionats, i s'anoten totes les adreces de les imatges en aquest arxiu. S'ha de comenta que es genera un arxiu input per a cada execució de mètode que es realitza.

Exemple d'arxiu input

```
f;/home/group/projecte/data/images/datasets/flickr/desert/3336142322_95f3be313e.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/3343433780_22fdd73b5d.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2841554573_b4ca106ca8.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2395610633_6252e21f90.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/3565972098_67fe9315f2.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2558143729_2b96b63cb1.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/434803937_fe3889840b.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/374886727_11607729c4.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2842605155_0d06415672.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2805343313_59760510ab.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2218530751_efef9e55ec.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/808912643_c031026930.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2807518123_e4d0ff9d74.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2599529153_63da0637bc.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/378311777_cba58150ee.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/245690441_b97ad57a8b.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/3064571796_edba55f6fa.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2676685319_8dc224ed60.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/2239285438_d459be9fb4.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/3222391757_de6a68a45d.jpg
f;/home/group/projecte/data/images/datasets/flickr/desert/1185840241_c57f17ba41.jpg
```

Com es pot observar cada línia del arxiu és una imatge que està continguda en algun dels directoris seleccionats. La primera opció de totes les línies pot

estar definida com a 'd' si es tracta d'un directori sencer, o com a 'f' si només es tracta de l'adreça d'una imatge.

- **arxius output** són iguals que els arxius input. Aquests arxius però són generats des de els mètodes, és a dir , un cop executat el procés els mètodes escriuen aquests arxius amb l'ordre amb que han de ser mostrades les imatges a la web. Trobem doncs totes les adreces de les imatges que han de ser mostrades amb un cert ordre. A més cada imatge pot contenir altres camps (números estadístics) per a ser mostrats a sota de la imatge. La primera fila dels fitxer *outputs* conté el nom de cada un d'aquests paràmetres associats a cada adreça d'imatge.

4.1.3. Creació de la web

En aquesta secció s'explicarà pas a pas quina ha sigut la construcció de l'aplicació web. Inicialment es farà un resum de quina es l'estructura de la web explicant el contingut de cada directori. En les altres seccions es mostrarà quines han sigut les tècniques d'implementació dels punts més importants de la web així com la visualització de cada un d'ells.

4.1.3.1. Estructura de la web

La web està estructurada en un conjunt de directoris els quals tenen tots la seva funció:

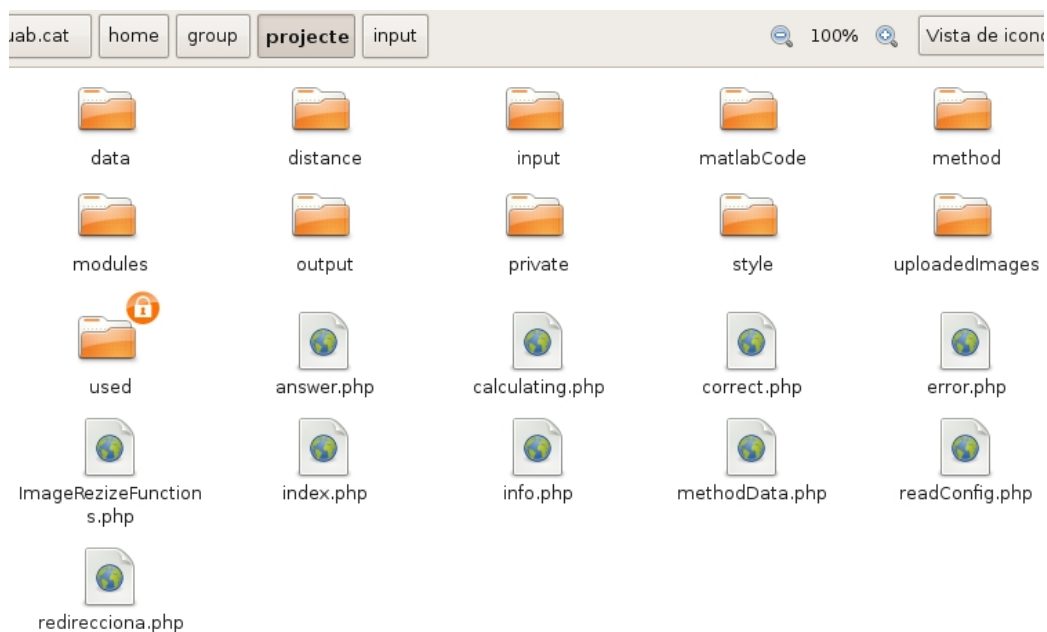


figura 20. Estructura base de directoris de la web.

- **data** , dins d'aquest directori aniran guardades totes les bases de dades d'imatges i altres característiques necessàries.

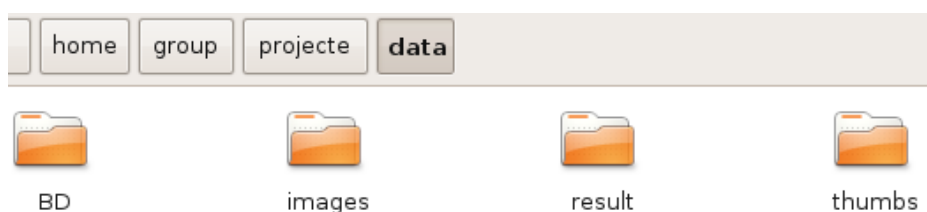


figura 21. Estructura del directori data de la web.

Això és el que tindrem dins d'aquest directori:

- **images/datasets** , directori on tindrem guardats tots els conjunts d'imatges amb les que es podran fer les consultes. Tindrem tants directoris com conjunts d'imatges diferents vulguem tenir i dins de cada un d'ells es podran definir altres sub directoris. Les bases de dades descarregades des de flickr estaran dins del directori flickr.

Amb aquesta estructura doncs podrem definir per exemple imatges de platja com a directori general (images/datasets/beach) i al seu interior definir dos sub directoris de nit i de dia (images/datasets/beach/night , images/datasets/beach/day).

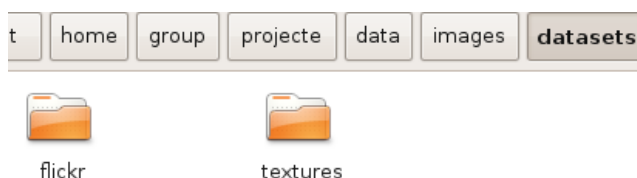


figura 22. Directori de images/datasets.

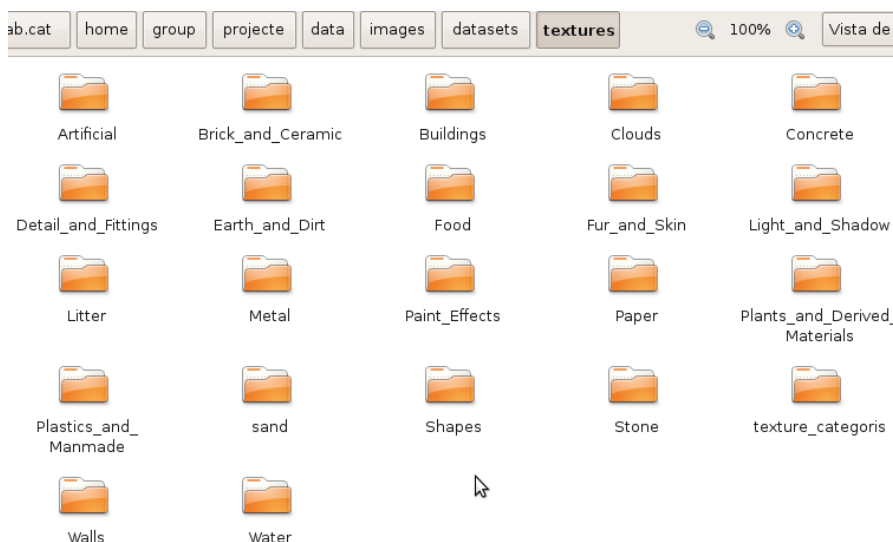


figura 23. Directori images/datasets/textures.

- **BD/ i result/**, aquests directoris seran utilitzats per als mètodes per a fer càlculs previs sobre les imatges, és a dir tots els mètodes per a poder treure resultats necessiten fer uns càlculs previs. El resultat de tots aquests càlculs previs de cada imatge seran guardats en aquests directoris. La forma de fer-ho serà mantenint la mateixa estructura que hi ha al “imatges/data sets”. D'aquesta manera quan el mètode vulgui fer algun càlcul d'alguna imatge podrà consultar si aquest ja esta guardat i per tan utilitzar-lo i estalviar-nos molt de temps.

Com que els mètode poden tenir moltes variants en els paràmetres

d'entrada i per tan els càlculs previs són diferents és guardarà un directori amb el nom del mètode i els paràmetres que s'han utilitzat, i dins seu tots els càlculs previs de les imatges consultades mantenint l'estructura comentada.

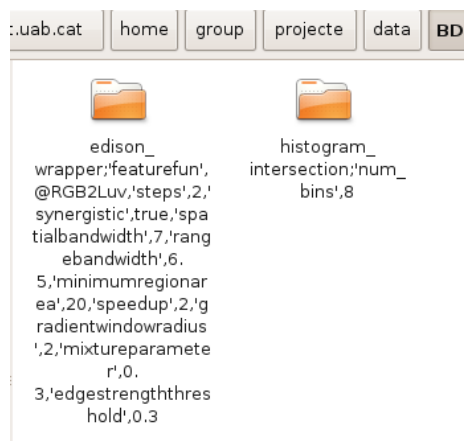


Figura 24. Dos directoris de càlculs previs amb el nom del mètode i els paràmetres.

- **thumbs/** Aquest directori té la mateixa estructura que el directori images/datasets. En ell es creen els thumbnails de les imatges que han de ser mostrades a la web. Els thumbnails només es creen una vegada els altres cops s'utilitza el thumbnail que ja està guardat. Si en algun moment es vol eliminar tots els thumbnails d'un dataset en concret serà molt fàcil, només caldrà buscar el mateix directori que tinguem a images/datasets però a thumb/.
- **distance/** conté l'arxiu distance.dis comentat en el capítol anterior (4.1.2).

- **input/ i output/** contenen es arxius que es generen amb les rutes de les imatges a analitzar comentat en el capítol anterior (4.1.2).

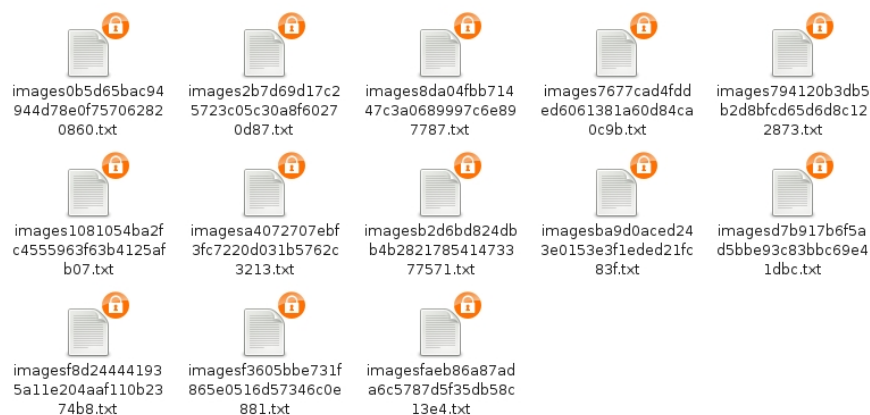


figura 25.Exemple de contingut del directori input.

- **matlabCode/** directori que conté tot el codi de la interfície matlab. També hi ha guardat el executable, que no és més que el codi compilat.
- **method/** directori que conté tots els arxius de configuració dels formularis dels mètodes, la configuració dels quals ha sigut comentada en l'anterior capítol (4.1.2). Cada fitxer que hi ha es la configuració d'un mètode diferent. Dins del directori també trobem un altre directori anomenat *description* en el qual hi han definides les descripcions de cada un dels mètodes definides en un arxiu per mètode (amb extensió .info).

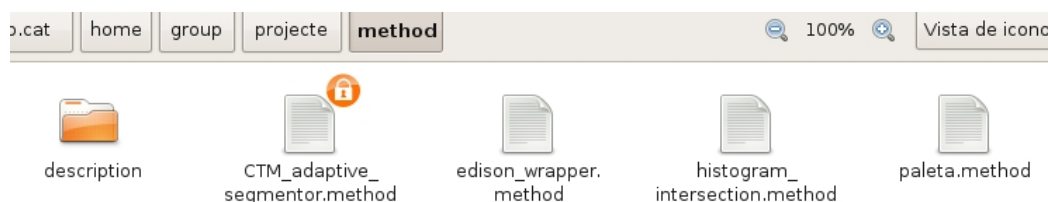


figura 26.Exemple de contingut del directori method.

- **modules/** directori que conté els components que han sigut creats per ser utilitzats en formularis web. És tot el codi que genera aquests components (comentats a la secció 4.1.2).

- **private/** directori que conté la part d'intranet de la web. És aquí doncs, on estan els directoris en que estan programats la configuració de les web i les consultes a flickr per a la posterior descàrrega d'imatges.
- **style/** directori que conté el arxiu de definició de l'estil de la web (style.css).
- **uploadedImages/** directori que conté totes les imatges que els usuaris carreguen al servidor per a ser provades amb mètodes i conjunts d'imatges. Aquestes imatges són les que es pugen amb el component que s'ha creat per fer la pujada de les imatges.
- **used/** directori que conté tots els scripts per a fer proves que s'han utilitzat.

4.1.3.2. Consulta de mètodes

Aquest és el procés en que se selecciona quin és el mètode que es vol executar i quin és el conjunt d'imatges amb el que es vol treballar. Després es genera el formulari del mètode (configurat en el seu arxiu .method), es modifiquen els paràmetres que es vulguin i s'executa el mètode. Es destaquen els següents punts en la implementació d'aquesta secció:

- **Creació del formulari inicial**

```
<form action="index.php" method="POST" target="_self" enctype="multipart/form-data" name="frmTree">
<?php
include('readConfig.php');
// Available extensions
$ext = array("jpg", "jpeg", "png","gif","JPG");

// Print the method combobox
echo "<div style='border-bottom: 1px dotted rgb(204, 204, 204); padding-bottom: 10px;'>";
echo "<img src='style/title.jpg' width='270' style='padding-right: 7px;'>";
pintaComboAplicacions("method/", "method");
?>
<a href="#" onclick='showInfo();'><img src='style/info.png' width='20' title='Open/Close information window'></a>

<?
echo "</div>";
// It calls to the print tree directory function
echo "<div style='border: 0px solid rgb(0, 0, 0);background:#EEE; overflow-x:
|auto;overflow-y:scroll; width: 500px; height:300px; margin-          top: 15px;'>";
echo tree($_SERVER['DOCUMENT_ROOT'].$dirProject."data/images", "javascript:alert('You clicked on [link]');",$ext);
echo "</div>";
echo "<br>";
echo "<div align='center'>";
echo "<input type='Submit' VALUE='Start method'></div><br>";

?>
</form>
```

figura 27. Codificació formulari inicial

- **Implementació del quadre de text en el qual hi han els mètodes a seleccionar.** Es fa a partir d'una funció que llegeix el nom de tots els arxius (configuració del formulari de cada mètode) continguts en el directori method/.

```
function pintaComboAplicacions($dir,$ext)
{
    echo " <select name='method'>";
    if($ds=opendir($dir)) {
        //Check all the files into the directoy. We have a file for each method.
        while(($arxiu=readdir($ds))!==false) {
            if(!is_dir($dir."/".$arxiu)and($arxiu!='.')and($arxiu!='..'))
            {
                $extensio = strrchr($dir."/".$arxiu,'.');
                $extensio = substr($extensio,1,strlen($extensio));
                if($extensio==$ext)
                {
                    //Print next value of method on the list
                    echo " <option value='". $dir.$arxiu."'>".substr($arxiu,0,strlen($arxiu)-7)."</option>";
                }
            }
        }
    }
    echo "</select>";
}
```

Figura 28. Funció quadre de text de mètodes.

- **Creació del arxiu input amb les adreces de les imatges pertanyents als conjunts d'imatges escollits.** Aquest directori es genera a partir d'una funció recursiva que mira a dins de cada directori i subdirector i anota les adreces de les imatges que hi troba. Cal tindre en compte que es pot escollir un directori superior i automàticament ja s'introduiran totes les imatges que també existeixin en els seus subdirectoris.

```

//Recursively function. It checks all the images on the directory and if exist in all of the subdirectoris
//This function needs: $dir:directory to be analyzed , $fitxer: file where it will put the images path
//
// $ext: permitted extensions
function examinarDir($dir,$fitxer,$ext) //$fitxerproves
{
    if($ds=opendir($dir)) {
        while(($arxiu=readdir($ds))!==false) {
            //If the current file is a directori it calls the function again
            if (is_dir($dir."/".$arxiu)and($arxiu!='.')and($arxiu!='..'))
            {
                examinarDir($dir."/".$arxiu,$fitxer,$ext);
            }
            //If the current file is a image file it writes the image path in the input file
            elseif(!is_dir($dir."/".$arxiu)and($arxiu!='.')and($arxiu!='..'))
            {
                //it checks if the file has a permitted extension
                $extensio = strrchr($dir."/".$arxiu,'.');
                $extensio = substr($extensio,1,strlen($extensio));
                if(in_array($extensio, $ext))
                {
                    //write the path in the input file
                    fputs($fitxer,"f;".$dir."/".$arxiu."\n");
                }
            }
        }
    }
}

```

Figura 29. Funció creació del arxiu input.

➤ **Visualització de la pantalla en que s'implementen els codis anteriors**



Figura 30. Formulari inicial de la web.

- **Creació dinàmica del formulari associat al mètode seleccionat.** Un cop s'ha seleccionat el mètode i el conjunt d'imatges es genera el formulari amb tots els paràmetres associats al mètode escollit. És la funció encarregada de examinar el arxíu .method i crear cada component del formulari tal i com estan definit en el arxíu.

```
function printFormulariMetode($metode,$inputArxíu,$outputArxíu,$root)
{
    $ds = fopen($metode,"r");
    //Read the first line of the file.
    $linia = fgetcsv ( $ds , 2048, "","'\'\' ');
    //Read the number of fields
    $entrades = $linia[0];

    //For each field, print the component
    for ($i=0;$i<=$entrades;$i++){
        $linia = fgetcsv ( $ds , 2048, "","'\'\' ');
        if($linia[0]=='UPLOAD')
        {
            //If the mthod has a upload component it makes 2 forms. The first one is for uploading an image(s).
            echo "<form action='/cgi-bin/uploadMul.cgi' method='POST' target='_self' enctype='multipart/form-data' name='frmEnvia'>";
            echo "<p>Image(s): <input type='file' name='photo' onChange='copyNameImage()'>";
            echo "<input name='location' type='hidden'><input name='locationAdd' type='hidden'><input type='button' VALUE='clear'
                onclick='deleteCookie()' ></p>"; //type='hidden'

            //add div for show charge bar, change the path if you need
            echo "<div id='wait' style='display:none' align='left'>";
            echo "Uploading image ...<br>";
            echo "<img src='modules/barra.gif'></div>";
            // add div for uploaded file
            echo "<div id='uploadImage' style='display:none'></div>"; //<img name='imageToUpload'>
            echo "</form>";
            //Second method
            //This method calls the matlab program with all the parameters.
            echo "<form method='get' action='/cgi-bin/webMatlab/matlabInterface' name='frmEnvia2'>";
            //Importan fields that it will be hidden. All the needed parameters : path of the input,
            //output files of images, path of the web, name anf file of the method
            echo "<input name='inputarxíu' SIZE='100' value='$inputArxíu' type='hidden'>"; //type='hidden'
            echo "<input name='outputarxíu' SIZE='100' value='$outputArxíu' type='hidden'>"; //type='hidden'

            echo "<input name='metode' SIZE='100' value='".substr($metode,7,strlen($metode)-(strlen(".method")+7))."' type='hidden'>"; // son 7 de
            method/ + 7 de .method //type='hidden'
            echo "<input name='metodearxíu' SIZE='100' value='/projecte/'. $metode.'" type='hidden'>"; //type='hidden'
            echo "<input name='$linia[1]' SIZE='50' type='hidden'><br>"; //type='hidden'
            echo printDistanc("distance/distance.dis");
            echo "<input name='root' SIZE='100' value='". $root.'" type='hidden'><br>"; //type='hidden'
        }
        //if the method doesn't has an upload field the form will be the same without the upload field ($linia[1]).
        else if(($linia[0]!='UPLOAD') and ($i==0))
        {
            echo "<form action='/cgi-bin/matlabInterface' method='get' target='_self' enctype='multipart/form-data' name='frmEnvia'>";
            echo "<input name='inputarxíu' SIZE='100' value='$inputArxíu' type='hidden'>"; //type='hidden'
            echo "<input name='outputarxíu' SIZE='100' value='$outputArxíu' type='hidden'>"; //type='hidden'
            echo "<input name='metode' SIZE='100' value='".substr($metode,7,strlen($metode)-(strlen(".method")+7))."' type='hidden'>"; // son
            7 de method/ + 7 de .method //type='hidden'
            echo "<input name='metodearxíu' SIZE='100' value='/projecte/'. $metode.'" type='hidden'><br>"; //type='hidden'
            echo printDistanc("distance/distance.dis");
            echo "<input name='root' SIZE='100' value='". $root.'" type='hidden'><br>"; //type='hidden'
        }
        //other components in the form. All the lines (in the file) after upload field.
        //Text component
        if($linia[0]=='TEXT')
        {
            echo "$linia[2] <input type='$linia[0]' name='$linia[1]' value='$linia[3]' size='$linia[4]'><br>";
        }
        //Checkbox component
        if($linia[0]=='CHECKBOX')
        {
            if ($linia[3]=='false')
            {
                echo $linia[2];
                echo " true<input type='radio' name='$linia[1]' value='true'>";
                echo " $linia[3]<input type='radio' name='$linia[1]' value='". $linia[3]."' checked disabled><br>"; //onclick='changeChecked(this)'
            }
        }
    }
}
```

```


    else
    {
        echo $linia[2];
        echo " $linia[3]<input type='radio' name='$linia[1]' value='". $linia[3]."' checked>";
        echo " false<input type='radio' name='$linia[1]' value='false' ><br>";
    }
}
//Combobox component
if($linia[0]=='COMBOBOX')
{
    echo $linia[2];
    echo " <select name='$linia[1]' >";
    for($element=4;$element<4+$linia[3];$element++)
    {
        echo " <option value='". $linia[$element]."'>". $linia[$element]. "</option>";
    }
    echo "</select><br>";
}
//Colour picker component
if($linia[0]=='PALETA')
{
    $indexColor = 2;
    echo "<div id='picker' style='float: left;'></div>";
    for($j=0;$j<$linia[1];$j++){
        echo "<div class='form-item'><label for='$linia[$indexColor]'" . $linia[$indexColor + 1]. "</label><input type='text' id='$linia["
        [$indexColor]' name='$linia[$indexColor]' class='colorwell' value='#123456' /></div>";
        $indexColor = $indexColor+2;
    }
}
}
}

```

Figura 31. Codificació de la creació dinàmica de formularis de mètodes.

➤ Visualització del formulari creat dinàmicament

Image(s): Examinar... clear



Distance:

featurefun:

steps:

synergistic: ☒ true ☐ false

SpatialBandWidth:

RangeBandWidth:

MinimumRegionArea:

SpeedUp:

GradientWindowRadius:

MixtureParameter:

EdgeStrengthThreshold:

run method

Figura 32. Formulari del mètode.

4.1.3.3 Execució de mètodes

La secció anterior s'acaba mostrant el formulari en el qual apareixen tots els paràmetres del mètode. Aquest formulari té com a acció la d'executar la interfície que ens comunicarà amb els mètodes matlab. Aquesta interfície rep tots els paràmetres continguts en el formulari i els prepara per a fer la crida de la funció *ImRetrieval* que és l'encarregada d'executar el mètode seleccionat. L'execució d'un mètode ha de tindre la següent nomenclatura:

```
imRetrieval('adreça_del_arxiu_input','nom_mètode;'nom_parametre",valor,..','adreça_
del_axiu_output','imatge','adreça_de_la_imatge_a_consultar','distancia','nom_distancia'
);
```

Un exemple de la crida a un mètode seria el següent:

```
imRetrieval('/home/group/projecte/projecte/file.txt','histogram_intersection;'num_bins'
',8','/home/group/projecte/output/fileOUT.txt','imatge','/home/group/uploadedImages/i
matge.jpg','distancia','euclidean');
```


➤ Implementació de la interfície matlab

```
function matlabInterface()
% input paramters of the form are in the environment variable "QUERY_STRING"
% user inform message. Calculating method..
fprintf(1,'Content-Type: text/html; charset="iso-8859-15 \n\n ');
s=urllread('http://www.cat.uab.cat/projecte/calculating.php');
fprintf(1,s);
input = qs2struct(getenv('QUERY_STRING'));% we get the input form parameters in 'input var'

try
% it saves the original parameter to send again in the web after te process
output.metodearxiu = input.metodearxiu;
output.inputarxiu = input.inputarxiu;
output.outputarxiu = input.outputarxiu;
output.outputarxiu = input.outputarxiu;
output.metode = input.metode;

input.root = replaceOnlyWebSlash(input.root);
input.metodearxiu = replaceWebSlash(input.metodearxiu,input.root);
input.inputarxiu = replaceWebSlash(input.inputarxiu,input.root);
input.outputarxiu = replaceWebSlash(input.outputarxiu,input.root);
try
    output.imatge = input.imatge;
    input.imatge = replaceWebpoint(input.imatge,input.root);
catch
end

%it gets all the names of the parameters
camps = fieldnames(input);
%All the fields between 2..8 are required
%it generates the method parameter
metode = [input.metode ''];
for i=9:numel(camps)
    field = getfield(input,camps{i});
    if (strcmp([' field '],'O')~=1)
        field = convertToMat([' field ']);
        field = replacearoba([' field ']);
        field = toRGB([' field ']);
    end

    if (strcmp([' camps{i} ','colormetric']))
        metode = [metode '' camps{i} '' field ''];
    else
        if (i~=numel(camps))
            metode = [metode '' camps{i} '' field ''];
        else
            metode = [metode '' camps{i} '' field ''];
        end
    end
end
end
%it put all the parameters in a var param
%input images file
param{1}=[' input.inputarxiu '];
%the method with all the parameters
param{2}=[' metode '];
%output image file
param{3}=[' input.outputarxiu '];
%if the image doesn't exist just put the distance parameter (name,value)
if (numel(input.imatge)==0)
```

```

    param{4}='distancia';
    param{5}=[' ' input.distancia ''];
else
%if the image exist it puts the image (name,path) and the distance (name,value) parameters
    param{4}='imatge';
    param{5}=[' ' input.imatge ''];
    param{6}='distancia';
    param{7}=[' ' input.distancia ''];
end
%crida contains the string with all the parameters to call the method process
crida = [param{:}];
%it calls the ImRetrieval function with the parameters. This function will call the method defined on param{2}
feval(@ImRetrieval,param{:});
%when the process is finished it calls the answer page to show all the results
try
    s=urldread(['http://www.cat.uab.cat/projecte/redirecciona.php?outputarxiu='
        output.outputarxiu '&metode=' output.metode '&metodearxiu=' output.metodearxiu '&imatge=' output.imatge]);
    fprintf(1,'la crida a la funcio: %s',crida);
catch
    s=urldread(['http://www.cat.uab.cat/projecte/redirecciona.php?outputarxiu='
        output.outputarxiu '&metode=' output.metode '&metodearxiu=' output.metodearxiu]);
end
fprintf(1,s);
%if there are some error it calls an error page
catch e
    s=urldread('http://www.cat.uab.cat/projecte/error.php');
    fprintf(1,s);
    fprintf(1,getReport(e,'extended'));
end
end

```

Figura 33. Codificació de la interfície matlab.

Després de l'execució del mètode es crida a la pàgina web de visualització dels resultats, associant totes les variables necessàries.

Aquesta web llegeix el arxiu output generat pel mètode i i genera la corresponent pantalla de visualització dels resultats. En aquest punt ens podem trobar amb dos possibles visualitzacions diferents:

- **Visualització de la pantalla de resultats si l'usuari ha introduït una imatge de consulta**


histogram_intersection


column

row-pag


Go

Consulted images







Distance:0.418265




Distance:0.419616




Distance:0.426126




Distance:0.435675




Distance:0.463184



Distance:0.465160



Distance:0.466605



Distance:0.476549

Figura 34. Pantalla de resultats de l'execució del mètode amb imatge de consulta.

➤ **Visualització de la pantalla de resultats si l'usuari no introdueix imatge de consulta**

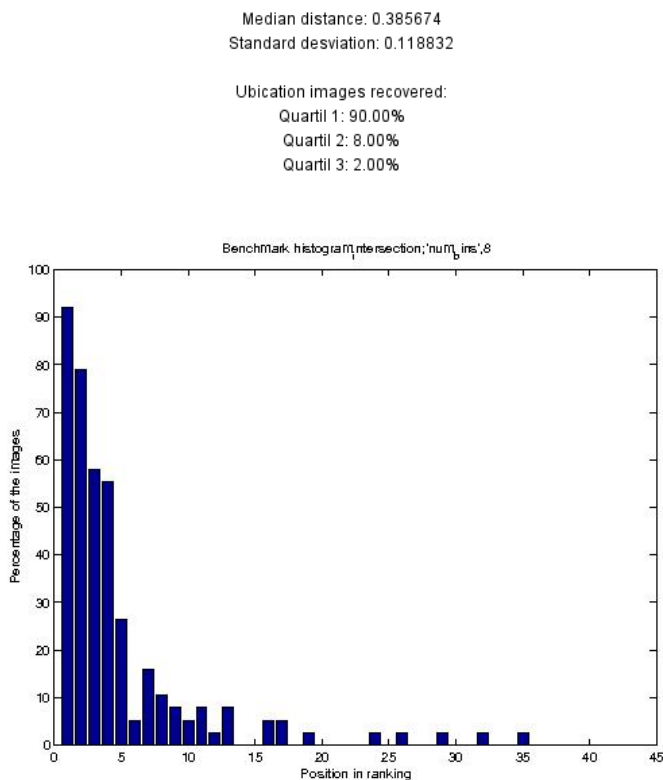


Figura 35. Pantalla de resultats de l'execució del mètode sense imatge de consulta.

4.1.3.4 Cercador d'imatges flickr

A continuació es comentaran les parts més importants en la implementació de la cerca i descàrrega d'imatges de flickr.

Inicialment es començar creant el formulari de consulta a les imatges de flickr. Aquest formulari és executat per filtrar el tipus d'imatges i així poder acotar la cerca.

➤ **Implementació del formulari de cerca**

[illegible]

Figura 36. Codificació del formulari de cerca a flickr.

➤ **Visualització del formulari de cerca**

☒ Tags ☒ or ☐ and ☐ Text descriptor

Images for page (max value 500)

Show tags ☐

Figura 37. Formulari de cerca a flickr.

Un cop s'executa la cerca, es passen tots els paràmetres de cerca a la pàgina encarregada de fer la connexió a flickr i extreure les imatges desitjades. Per a fer aquesta connexió es necessita una clau ID. Amb aquesta clau podem crear un objecte de la API de flickr per poder fer la connexió amb les seves imatges.

- **Creació del objecte i connexió a flickr** ,a la segona línia de codi es crea el objecte de la API de flickr i a la ultima línia es fa la connexió filtrant per totes les dades recuperades del formulari de consulta a flickr (comentat en el punt anterior).

```
//the phpFlickr Api is required
require_once("phpFlickr.php");
//It creates a new connection to flickr with a ID
$f = new phpFlickr("d51a852e16430fb26b78b3435cb48df0");
//It gets all the parameters that comes from the form
if (isset($_POST['busqueda']))
    $tags = $_POST['busqueda'];
if (isset($_POST['pg']))
    $pg = $_POST['pg'];
if (isset($_POST['tagMode']))
    $tagMode = $_POST['tagMode'];
if (isset($_POST['perPage']))
    $perPage = $_POST['perPage'];
if (isset($_POST['geoContext']))
    $geoContext = $_POST['geoContext'];
if ($_POST['busquedatex'] != '')
    $busquedatex = $_POST['busquedatex'];
//it makes the query to the flickr images
$photos = $f->photos_search(array("tags"=>$tags,"tag_mode"=>$tagMode,"sort"=>
    "interestingness-desc", "privacy_filter"=>"1", "per_page"=>$perPage,
    "page"=>$pg, "text"=>$busquedatex,"media"=>"photos"));
```

Figura 38. Codificació de la connexió a flickr.

- **Creació de la graella d'imatges trobades** , un cop es rep la resposta de flickr, la variable \$photos contindrà totes les imatges trobades amb les característiques demanades. A partir d'aquesta variable es van mostrant totes les imatges per pantalla.

```

if (is_array($photos['photo']))
{
    $columnes = 0;
    echo "<table cellpadding='12px'><tr>";
    foreach ($photos['photo'] as $photo)
    {
        //we want to recover the square size (thumbnail for show) and the large size ( image for download )
        $rutaPetita = $f->buildPhotoURL($photo, "square");
        $rutaMesGran = $f->buildPhotoURL($photo, "large");
        echo "<td>";
        $columnes++;
        $str = "";
        //If we want to see the tags (previously selected on the form). It recover all the tags and puts
        // in the alt and title of the image recovered
        if($_POST['mostraTags']==1)
        {
            $imTags = $f->tags_getListPhoto($photo['id']);
            foreach ($imTags as $tag)
            {
                $str.=$tag['raw']. " | ";
            }
        }
        //here it print the image
        $salida = "<div class='caja'>";
        $salida .= "<a href='".$rutaMesGran."'><img alt='".$str."' title='".
            . $str."' src='".$rutaPetita."' /></a><br>";
        echo $salida."</div></td>";

        //take a look of the maximum number of columns
        if ($columnes==12)
        {
            echo "</tr><tr>";
            $columnes=0;
        }
    }
}

```

Figura 39. Codificació de la creació de la graella d'imatges.

- **Visualització de les imatges trobades a flickr**

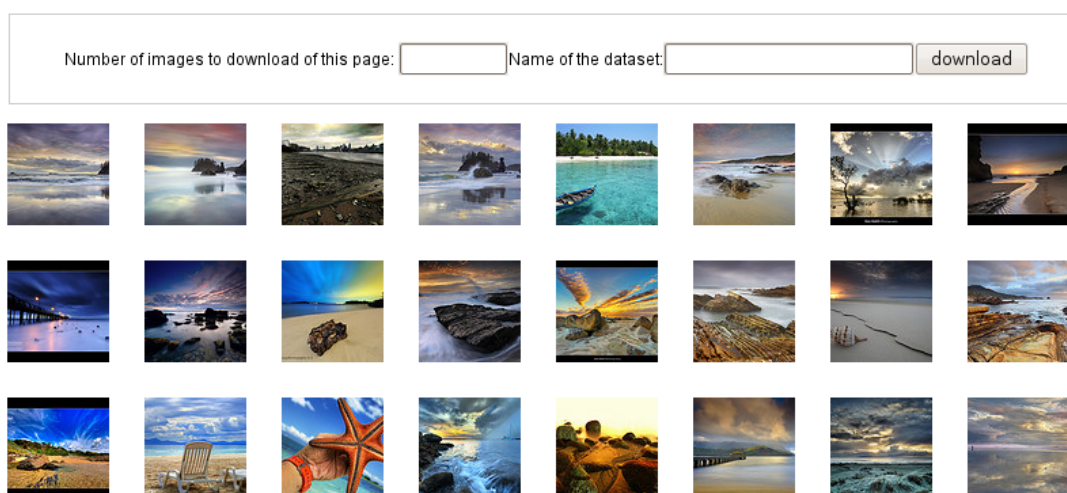


Figura 40. Pantalla de visualització de les imatges de flickr.

El formulari que apareix sobre de les imatges trobades és l'encarregat d'executar el procés de descàrrega de les imatges que interessin al usuari administrador. És en aquest punt, doncs s'ha implementat una funció en la que es descarreguen les imatges a un directori (amb el nom que el usuari administrador tria) del servidor web.

➤ Implementació de la funció descàrrega de les imatges de flickr

```
//It recovers the images in the page selected ($photos)
$photos = $f->photos_search(array("tags"=>$tags,"tag_mode"=>$tagMode,"sort"=> "interestingness-desc",
"privacy_filter"=>"1", "per_page"=>$perPage,"page"=>$pg, "text"=>$busquedatex,"media"=>"photos"));
//It moves for all the image. It will download the number of images writed in $numIm
while(($imatges<$numIm)and(($imatges<$photos['total'])))
{
    //it charges the images in the page that we was $pg
    $photos = $f->photos_search(array("tags"=>$tags,"tag_mode"=>$tagMode,"sort"=> "interestingness-desc",
    "privacy_filter"=>"1", "per_page"=>$perPage, "page"=>$pg, "text"=>$busquedatex,"media"=>"photos"));
    //add one page for the next iteration (if isn't complet the number of images to download)
    $pg++;
    //it save each photo on $photo
    foreach ($photos['photo'] as $photo)
    {
        //it checks if it lefts images to download
        if(($imatges<$numIm))
        {
            //saves the path of the large image
            $rutaMesGran = $f->buildPhotoURL($photo, "large");
            //it checks if the large image exists
            if(verificar_url($rutaMesGran))
            {
                //if exists it adds a new one image in $imatges
                $imatges++;
                //It copies the image in the server. In the directory named in the form.
                $m = fopen($ruta.$photo['id']."_".$photo['secret'].'.jpg','w+');
                $im = file_get_contents($rutaMesGran);
                fwrite($m,$im);
                fclose($m);
            }
        }
    }
}
```

Figura 41. Codificació de la funció de descàrrega.

4.1.3.5 Configuració web

En aquesta secció s'explica la implementació del procés en que els usuaris administradors poden configurar els paràmetres de configuració de la web. Així doncs, el que inicialment es necessita saber és on estan guardats aquests paràmetres.

➤ **Arxiu de configuració de la web config.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<!-- All the main configuration of the interface -->
<config>
  <directory>
    <images>data/images/datasets/</images>
    <thumb>data/thumb/</thumb>
    <flickr>private/flickr/</flickr>
    <project>/projecte/</project>
    <output>output/</output>
    <input>input/</input>
    <method>method/</method>
    <modules>modules/</modules>
    <distance>distance/</distance>
  </directory>
  <thumb>
    <pheight>400</pheight>
  </thumb>
</config>
```

Figura 42. Contingut del arxiu config.xml.

Es pot observar com el primer conjunt de paràmetres són tots els directoris de la web (*directory*) , mentres que el segon són els paràmetres de configuració de la creació de thumbnails (*thumb*).

Per a la lectura i escriptura d'aquest arxiu s'han implementat dues funcions:

➤ **Funció de lectura del arxiu config.xml**

```
//create a new DOM component to read the xml file
$doc = new DOMDocument();
$doc->load( 'config.xml' );
//read all the fields of the file
//save it in variables
//directory options
$directorys = $doc->getElementsByTagName( "directory" );
foreach( $directorys as $directory )
{
    $images = $directory->getElementsByTagName( "images" );
    $dirImages = $images->item(0)->nodeValue;

    $thumbs = $directory->getElementsByTagName( "thumb" );
    $dirThumbs = $thumbs->item(0)->nodeValue;

    $flickr = $directory->getElementsByTagName( "flickr" );
    $dirFlickr = $flickr->item(0)->nodeValue;

    $project = $directory->getElementsByTagName( "project" );
    $dirProject = $project->item(0)->nodeValue;

    $output = $directory->getElementsByTagName( "output" );
    $dirOutput = $output->item(0)->nodeValue;

    $input = $directory->getElementsByTagName( "input" );
    $dirInput = $input->item(0)->nodeValue;

    $method = $directory->getElementsByTagName( "method" );
    $dirMethod = $method->item(0)->nodeValue;

    $modules = $directory->getElementsByTagName( "modules" );
    $dirModules = $modules->item(0)->nodeValue;
    $distance = $directory->getElementsByTagName( "distance" );
    $dirDistance = $distance->item(0)->nodeValue;
}
//thumbnail options
$thumbnails = $doc->getElementsByTagName( "thumb" );
foreach( $thumbnails as $thumbPropieties )
{
    $pheight = $thumbPropieties->getElementsByTagName( "pheight" );
    $heightThumb = $pheight->item(0)->nodeValue;
}
```

Figura 43. Funció de lectura del arxiu config.xml.

➤ Funció d'escriptura al arxiu config.xml

```

$doc->load( 'config.xml' );
//for all the items in the xml file
//it puts the field filled in the form configuration
$directorys = $doc->getElementsByTagName( "directory" );
foreach( $directorys as $directory )
{
    $images = $directory->getElementsByTagName( "images" );
    $images->item(0)->nodeValue = $_POST['dirImages'];

    $thumbs = $directory->getElementsByTagName( "thumb" );
    $thumbs->item(0)->nodeValue = $_POST['dirThumbs'];

    $flickr = $directory->getElementsByTagName( "flickr" );
    $flickr->item(0)->nodeValue = $_POST['dirFlickr'];

    $project = $directory->getElementsByTagName( "project" );
    $project->item(0)->nodeValue = $_POST['dirProject'];

    $output = $directory->getElementsByTagName( "output" );
    $output->item(0)->nodeValue = $_POST['dirOutput'];

    $input = $directory->getElementsByTagName( "input" );
    $input->item(0)->nodeValue = $_POST['dirInput'];

    $method = $directory->getElementsByTagName( "method" );
    $method->item(0)->nodeValue = $_POST['dirMethod'];

    $modules = $directory->getElementsByTagName( "modules" );
    $modules->item(0)->nodeValue = $_POST['dirModules'];

}
$thumbnails = $doc->getElementsByTagName( "thumb" );
foreach( $thumbnails as $thumbPropieties )
{
    $pheight = $thumbPropieties->getElementsByTagName( "pheight" );
    $pheight->item(0)->nodeValue = $_POST['heightThumb'];
}
//it saves the new configuration in the xml file
$doc->save('config.xml');

```

Figura 44. Funció d'escriptura del arxiu config.xml.

Per últim es mostrarà com és la pantalla en que es configura aquest arxiu:

➤ **Visualització de la pantalla de configuració de la web**

Configuration the directories of the aplicacion

Project directory

Dataset images directory

Thumbnails directory

Flickr directory

Output directory

Input directory

Method directory

Modules directory

Distances directory

Thumbnails propieties

Height of the thumbnail

Clean the old thumbnails ☐

Figura 45. Formulari de configuració de la web.

4.1.3.6 Control d'errors

L'aplicació ha sigut creada de tal manera que les dades sempre són correctes si es deixen les que dóna per defecte. En alguns casos però s'ha de controlar si s'han introduït totes. Per tots aquests casos s'han creat dues pàgines en les que s'informa a l'usuari si hi ha hagut algun error (i quin ha sigut) o si pel contrari l'operació s'ha processat correctament.

- **Exemple de control d'error en la descàrrega d'imatges de flickr** , donaria error en el cas que el usuari administrador no omplis el paràmetre del nom del directori on es volen guardar les imatges descarregades.

```
if(($numIm=='')or($_POST['nomDS']=='')){
    $message = "download the flickr images, you need to put a number of images and the name of the dataset";
    echo "<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0\";URL=../../error.php?text=\".$message.\">";
}
```

Figura 46. Control d'error per falta de dades.

Com es pot observar si les dades són buides l'aplicació executa una pàgina d'error amb el missatge d'error passat per paràmetre (guardat a la variable \$message). Tots els controls d'errors de la web són del mateix estil.

➤ Visualització del error



Figura 47. Visualització del error per pantalla.

Cal comentar que els missatges d'operació correcte segueixen la mateixa configuració, és a dir hi ha una única pàgina que mostra aquests missatges i el missatge que mostra es una variable que se li passa a la pàgina. D'aquesta manera podem enviar missatges diferents depenent de quin és el procés que s'analitza.

4.1.3.7 Fragments de codi a destacar

A continuació s'explicarà la implementació d'altres fragments de codi importants pel treball que ha suposat la seva programació. Ens estem referint a la programació dels components per a formulari que s'han creat.

➤ Implementació de l'arbre de directoris

```
function tree($directory, $return_link, $extensions = array()) {
    // Generates a valid XHTML list of all directories, sub-directories, and files in $directory
    // Remove trailing slash
    if( substr($directory, -1) == "/" ) $directory = substr($directory, 0, strlen($directory) - 1);
    $code .= php_file_tree_dir($directory, $return_link, $extensions);
    return $code;
}

function php_file_tree_dir($directory, $return_link, $extensions = array(), $first_call = true) {
    // Recursive function called by php_file_tree() to list directories/files
    // Get and sort directories/files
    if( function_exists("scandir") ) $file = scandir($directory); else $file = php4_scandir($directory);
    natcasesort($file);
    // Make directories first
    $files = $dirs = array();
    foreach($file as $this_file) {
        if( is_dir("$directory/$this_file" ) ) $dirs[] = $this_file; else $files[] = $this_file;
    }
    $file = array_merge($dirs, $files);

    // Filter unwanted extensions
    if( !empty($extensions) ) {
        foreach( array_keys($file) as $key ) {
            if( !is_dir("$directory/$file[$key]") ) {
                $ext = substr($file[$key], strrpos($file[$key], ".") + 1);
                if( !in_array($ext, $extensions) ) unset($file[$key]);
            }
        }
    }

    if( count($file) > 2 ) { // Use 2 instead of 0 to account for . and .. "directories"
        $tree = "<ul>";
        if( $first_call ) { $tree .= " class=\"php-file-tree\""; $first_call = false; }
        $tree .= ">";
        foreach( $file as $this_file ) {
            if( $this_file != "." && $this_file != ".." ) {
                if( is_dir("$directory/$this_file") ) {
                    // Directory
                    $tree .= "<li class=\"pft-directory\"><input
                        type='checkbox' name='\$directory/\$this_file'><a href='\"#\"'> .
                        htmlspecialchars($this_file) . \"</a>\";
                    $tree .= php_file_tree_dir(\"$directory/\$this_file\",
                        $return_link, $extensions, false);
                    $tree .= "</li>";
                } else {
                    // File
                    // Get extension (prepend 'ext-' to prevent invalid classes
                    // from extensions that begin with numbers)
                    $ext = "ext-" . substr($this_file, strrpos($this_file, ".") + 1);
                    $link = str_replace("[link]", "$directory/" . urlencode($this_file), $return_link);
                    $tree .= "<li class=\"pft-file \" . strtolower($ext) . \">
                        <a href='\"$link\"'> . htmlspecialchars($this_file) .
                        \"</a></li>";
                }
            }
        }
        $tree .= "</ul>";
    }
    return $tree;
}
```

Figura 48. Implementació arbre de directoris.

- **Implementació del paquet de funcions per a pujar una imatge al servidor** , per a fer el procés de pujada d'una imatge s'utilitza un script programat en perl. Aquest script és cridat des de el component de formulari encarregat de fer la selecció de la imatge a pujar. Quan el script guarda la imatge al servidor, també guarda una cookie. Així quan tornem a carregar el component del formulari es llegeix la cookie i es poden anar mostrant les imatges que es van pujant (així l'usuari sap que les imatges ja són guardades). Doncs bé, aquest paquet de funcions programat en javascript serveix per tot el tractament de les cookies (càrrega , lectura i eliminació de cookies).

```

function copyNameImage(){
    var sPath = location.pathname;
    var sPage = sPath.substring(0,sPath.lastIndexOf('/') + 1);
    document.forms[0].location.value = sPage + "uploadedImages";
    document.forms[0].locationAdd.value = document.forms[1].imatge.value;
    document.forms[0].submit();
    document.getElementById('wait').style.display = 'block';
}
function chargeCookie() {
    cookie = readCookie('pathImage');
    if (cookie)
    {
        var direccio=webAdress(punticoma(cookie));
        var vect = direccio.split(";");
        for (i=0; i<vect.length-1; i++){
            document.getElementById('uploadImage').innerHTML=
                document.getElementById('uploadImage').innerHTML + " ";
        }
        document.forms[1].imatge.value = webAdress(punticoma(cookie));
        document.getElementById('uploadImage').style.display = 'block';
        document.getElementById('wait').style.display = 'none';
    }
    else document.forms[1].imatge.value='';
}
function readCookie(nombre) {
    var ini = document.cookie.indexOf(nombre);
    if(ini == -1) return "";
    a = document.cookie.substring(document.cookie.indexOf(nombre + '=') +
        nombre.length + 1,document.cookie.length);
    return a;
}

function deleteCookie(){
    document.forms[1].imatge.value = '';
    document.forms[0].locationAdd.value = '';
    document.forms[0].location.value = '';
    document.forms[0].submit();
}
function webAdress(adress){
    var newAdress="";
    var vect = adress.split('%2F');
    for (i=0; i<vect.length; i++){
        newAdress = newAdress + "/" + vect[i];
    }
    return newAdress.substr(1,newAdress.length);
}

function punticoma(adress){
    var newAdress="";
    var vect = adress.split('%3B');
    for (i=0; i<vect.length; i++){
        newAdress = newAdress + ";" + vect[i];
    }
    return newAdress.substr(1,newAdress.length);
}

```

Figura 49. Paquet de funcions de suport a la pujada d'imatges al servidor.

- **Implementació del script perl que puja les imatges al servidor**, és el script comentat en el punt anterior. El que fa es rebre l'adreça de la imatge que ha de guardar, la guarda i després crea una cookie amb l'adreça del servidor on ha sigut guardada. En el cas en que es pugi més d'una imatge aquesta cookie conté totes les adreces de totes les imatges pujades separades per un punt i coma. El seu nom és uploadMulti.cgi i està guardat al directori cgi del servidor on també hi tenim guardats la compilació de la interfície de matlab amb tots els mètodes que poden ser consultats des de la web.

```
#!/usr/bin/perl -w

use strict;
use CGI;
use CGI::Carp qw ( fatalToBrowser );
use CGI::Cookie;
use File::Basename;

$CGI::POST_MAX = 1024 * 5000;
my $safe_filename_characters = "a-zA-Z0-9_.-";

my $query = new CGI;
my $filename = $query->param("photo");
my $location = $query->param("location");
my $locationAdd = $query->param("locationAdd");

my $pathim = "/Public".$location;
my $upload_dir = $ENV{DOCUMENT_ROOT}.$location;

if($location){
    if (-d $upload_dir){
        $upload_dir = $ENV{DOCUMENT_ROOT}.$location;
        $pathim = $location;
    }
    else{
        $upload_dir = $ENV{DOCUMENT_ROOT}.$pathim;
    }
}
if ( !$filename )
{
    print $query->header ( );
    print "There was a problem uploading your photo (try a smaller file).";
    exit;
}
```

```

my ( $name, $path, $extension ) = fileparse ( $filename, '\..*' );
$filename = $name . $extension;
$filename =~ tr/ /_/;
$filename =~ s/[^$safe_filename_characters]//g;

if ( $filename =~ /^[^$safe_filename_characters+]$/ )
{
    $filename = $1;
}
else
{
    die "Filename contains invalid characters";
}

my $upload_filehandle = $query->upload("photo");

open ( UPLOADFILE, ">$upload_dir/$filename" ) or die "$!";
binmode UPLOADFILE;

while ( <$upload_filehandle> )
{
    print UPLOADFILE;
}

close UPLOADFILE;

my $webpath = $ENV{HTTP_REFERER};
my $cookie = $query->cookie(-name=>'pathImage',-value=>
    $locationAdd.$pathim."/".$filename.";",-expires=>'+20m');
print $query->redirect(-cookie=>$cookie,-uri=>$webpath);
}
else{
my $webpath = $ENV{HTTP_REFERER};
my $cookie = $query->cookie(-name=>'pathImage',-value=>"" ,-expires=>'+1s');
print $query->redirect(-cookie=>$cookie,-uri=>$webpath);
}

```

Figura 50. Codificació perl del script que puja imatges al servidor.

4.1.4 Aplicació d'un estil

Aquest és l'últim apartat en el procés de creació de la web. Un cop s'ha aconseguit que totes les funcionalitats demanades, estiguin aplicades i en funcionament a l'aplicació web, és el moment de fer que la visualització de la web sigui més amigable i intuïtiva per al usuari. És en aquest punt en que es crea el arxiu de definició d'estil de la pàgina. Aquest és el arxiu *style.css* , que com s'indica en el seu nom utilitza una codificació css. En aquest arxiu es defineix com seran totes les parts de la web : colors , tamany i colors de les fonts, organització, marges, ... Tot definit en un sol arxiu. Les ventatges són clares, cada cop que es vulgui canviar algun aspecte de l'estil només s'haurà de tocar aquest arxiu i no una a una totes les pàgines que formen l'aplicació.

A cada pàgina s'ha de cridar l'arxiu estil perquè el tingui en compte, es fa de la següent manera:

```
<link href="http://www.cat.uab.cat/projecte/style/style.css" rel="stylesheet"
      type="text/css" />
```

Per a explicar aquesta com s'ha fet la definició d'estils s'ha dividit l'argumentació en totes les parts en que s'ha definit:

- **Creació del logotip** , s'ha creat el logotip amb la intenció que veient-lo quedi definit el que es fa en aquesta aplicació web. Es tracta d'una aplicació en que es poden fer demostracions de mètodes basats en *Image Retrieval* així doncs el logo és el següent:



Figura 51. Logotip de la web.

- **Definicions generals d'estil** ,són tots els estils que seran aplicats a totes les pàgines de la web.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    font-style: normal;
    margin-right:auto;
    margin-left:auto;
    margin:0 auto;
    text-align:justify;
}
img {
    border: 0 none;
}
a {
    color: white;
    text-decoration: none;
}
a:hover{
    color:white;
    text-decoration: underline;
}
```

Figura 52. Definicions generals d'estil.

Com es pot observar es defineixen els estils que s'aplicaran al *body* , els estils que s'aplicaran a la visualització de totes les imatges i també els estils que s'aplicaran als enllaços (per a totes les planes).

- **Definició de la capçalera de la web** , la capçalera conté tot el menú d'enllaços disponibles per l'aplicació web. La definició d'estils en el arxiu css és la següent:

```
#headerweb
{
    float: left;
    position: relative;
    display: block;
    background-image: url('header.png');
    background-repeat: repeat-x;
    color: white;
    height: 25px;
    width: 100%;
}
#SubHeaderweb
{
    position: relative;
    margin-right: 10px;
    margin-left: 30px;
    width: 98%;
}
```

Figura 53. Definició de l'estil de la capçalera.

La capçalera està formada per una part en la que es defineix la posició i les dimensions (*headerweb*) i per una altre en la que es defineix la posició de tot el que s'hi vulgui escriure (*subheaderweb*).

La seva visualització és la següent:



Optimal resolution 1280x1024 Home | Flickr query | Configuration

Figura 54. Visualització de la capçalera.

- **Definició del cos de la web** , el cos serà tota la part on es mostraran les dades. Aquestes dades poden ser les visualitzacions d'imatges com també la visualització de formularis. Cada un dels formularis mostrats també té el seu propi estil.

```
#bodyClass
{
  float:left;
  position:relative;
  display: block;
  width : 100%;
  text-align:justify;
  line-height: 18px;
  height:100%;
}
```

Figura 55. Definició de l'estil del cos de la web.

Com es pot observar a part de definir quines són les dimensions de la divisió, es defineix quina és la justificació del text o quin és l'espai entre línies.

- **Definició de formularis** , tots els formularis estan situats dins de la divisió del cos i per tan hereten totes les seves característiques. A més, però, també tenen la seva pròpia definició en el arxiu d'estil (style.css). Es veuran dos exemples de definició de formulari, el formulari associat a un mètode i el formulari de consulta a flickr.

```
#formulariFlickr
{
  margin: 0 auto;
  width: 1100px;
  border : 1px solid #CCC;
  padding : 20px 40px 20px;
  margin-top:20px;
}
#formulariMethod
{
  margin: 0 auto;
  width: 500px;
  height: 400px;
  border : 1px solid #CCC;
  padding : 20px 40px 20px;
  margin-top:20px;
  overflow-y: scroll;
}
```

Figura 56. Definició de l'estil dels formularis de la web.

La visualització d'aquests dos formularis s'ha vist en el punt 4.1.3.2.

- **Definició del peu de la web** , aquesta divisió sempre estarà situada a sota del tot de la web. Com que els tipus de pantalles poden canviar segons el que s'hagi de visualitzar s'han creat dos tipus de peu.
- **Peu amb posició variable**, s'utilitzarà a les pantalles en que es mostren imatges i el peu de la pàgina pot estar en una posició variable, és a dir , en la pantalla de visualització de resultats de l'execució de mètodes i a la pantalla de mostra de resultats de les consultes a flickr. Aquest peu sempre va al final de totes les imatges que es mostren. A més mostra un conjunt d'enllaços numerats que defineixen el paginat de totes les imatges que hi han per a visualitzar.

```
#footerwebMove
{
  float:left;
  position:relative;
  display: block;
  text-align:center;
  width:100%;
  height:30px;
  margin: 0 auto;
  background-image: url('footer.png');
  background-repeat: repeat-x;
  color: white;
  margin-top:20px;
}
```

Figura 57. Definició de l'estil del peu de la web amb posició variable.

- **Visualització del peu amb posició variable**



Figura 58. Visualització del peu de la web amb posició variable.

- **Peu amb posició fixe** , S'utilitza a la resta de pantalles a les que no es mostren imatge, normalment a totes les visualitzacions en que apareixen formularis. En aquest peu s'hi mostren un conjunt d'enllaços d'interès.

```
#footerweb
{
  float:left;
  position:absolute;
  display: block;
  text-align:center;
  width:100%;
  height:30px;
  margin: 0 auto;
  background-image: url('footer.png');
  background-repeat: repeat-x;
  color: white;
  margin-top:20px;
  bottom : 0;
}
```

Figura 59. Definició de l'estil del peu de la web amb posició fixe.

- **Visualització del peu amb posició fixe**



Contact webmaster | Colour and texture group | Computer Vision Center

Figura 60. Visualització del peu de la web amb posició fixe.

- **Implementació de l'estil en la codificació** , per poder aplicar un estil a una zona de la web, només cal emmarcar la zona de codi que es vol dividir etiquetant-la amb el nom en que tenim les divisions (vistes anteriorment) definides en el arxiu style.css. Així doncs per exemple si volem fer que una part del codi sigui la capçalera de la web s'ha fet el següent:

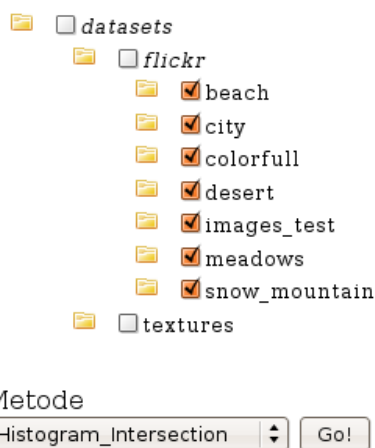
```
<div id='headerweb'>
<div id='SubHeaderweb'>
  <table><tr><td width='1100'>Optimal resolution 1280x1024</td><td width='200'>
    <a href='./'>Home</a> | <a href='private/flickr'>Flickr query</a> |
    <a href='private/configuration/config.php'>Configuration</a></td></tr></table>
  </div>
</div>
```

Figura 61. Implementació de l'estil en la codificació.

Com es pot observar en l'anterior imatges s'han posat dos <div> amb un id. Aquest nom d'id coincideix amb el nom que tenim definit al css, per tan tot el que estigui entre el <div> i </div> tindrà les característiques de capçalera de la web. Passa el mateix amb la sub capçalera definida al interior de la capçalera.

Per acabar la secció es mostraran dos exemples de quin ha sigut l'evolució de la web després d'aplicar l'estil. Es mostraran dues imatges per exemple, una de com era abans i l'altre com és ara (amb l'estil aplicat).

➤ **Formulari inicial abans d'aplicar l'estil**



datasets

- flickr
 - ☒ beach
 - ☒ city
 - ☒ colorfull
 - ☒ desert
 - ☒ images_test
 - ☒ meadows
 - ☒ snow_mountain
- textures

Metode

Histogram_Intersection Go!

Figura 62. Formulari inicial abans d'aplicar l'estil.

➤ **Formulari inicial després d'aplicar l'estil**



Figura 63. Formulari inicial després d'aplicar l'estil.

➤ **Visualització d'imatges després de l'execució d'un mètode. Abans d'aplicar l'estil.**

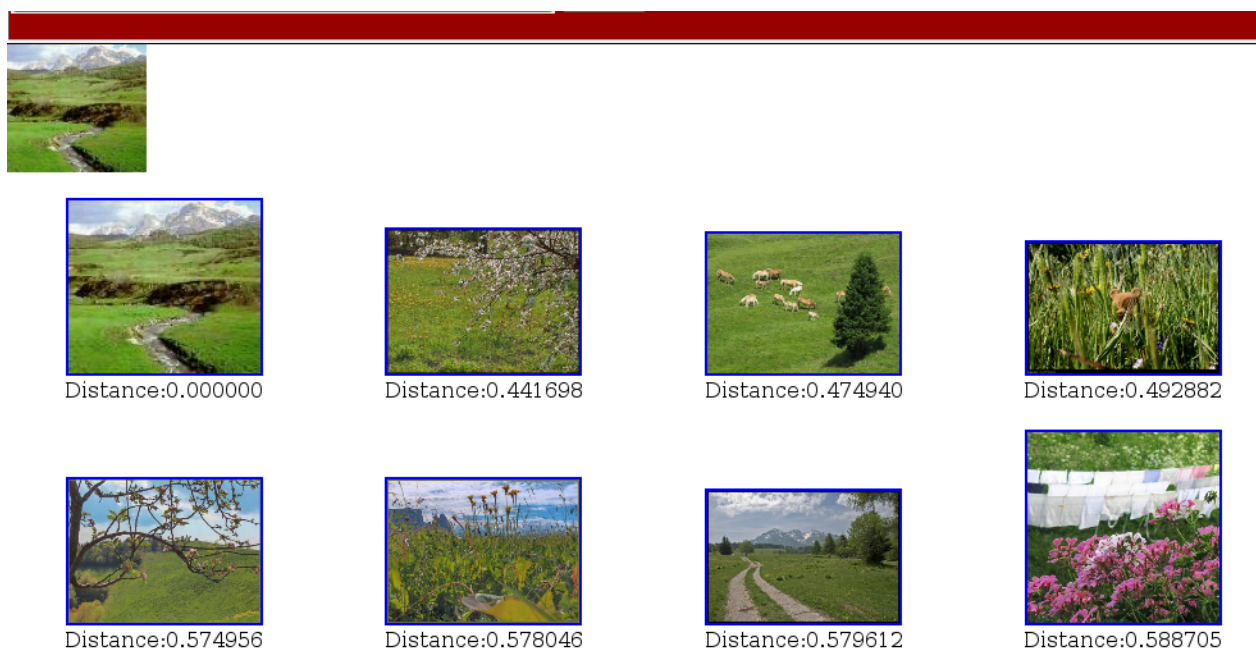



Figura 64. Visualització de resultats abans d'aplicar l'estil.

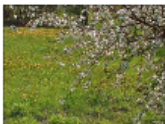
- **Visualització d'imatges després de l'execució d'un mètode. Després d'aplicar l'estil.**

histogram_intersection


column row-pag

Consulted images







Distance:0.441698




Distance:0.474940




Distance:0.492882



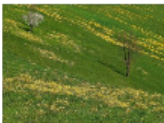
Distance:0.523541




Distance:0.579612




Distance:0.588705




Distance:0.591810




Distance:0.592878




Distance:0.609963



Distance:0.612399



Distance:0.612659



Distance:0.616493

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 Siguiente »

Figura 65. Visualització de resultats després d'aplicar l'estil.

4.2 Proves d'integració

Al llarg de tot el desenvolupament de l'aplicació web, s'han anat fent diferents proves per tal de verificar el correcte funcionament d'aquesta. En aquestes proves ens hem trobat amb els següents problemes que hem anat solucionant a mida que anaven apareixent. Cal dir que es fa un resum dels problemes més importants.

- El **pas de paràmetres entre l'aplicació web i els mètodes matlab** ha sigut un problema en quant al tipus de les dades. Inicialment els paràmetres es passaven tots com a tipus text i amb les cometes pertinents, però donava problemes amb els camps numèrics i *booleans*. Finalment s'ha decidit que totes les dades enviades des de el formulari del mètode seran de tipus text però sense enviar les cometes. D'aquesta manera els mètodes matlab llegeixen aquests paràmetres directament amb el tipus que esperen.
- En **l'execució de la interfície matlab** encarregada d'executar els mètodes, ens em trobat amb el problema de depuració d'errors. Al ser un programa compilat el que tenim és un executable al qual se'l crida i retorna una resposta, però en el cas que doni error no es pot saber en quina línia falla. Per solucionar aquest problema (fins aconseguir el correcte funcionament) s'ha anat fent que envies un missatge per pantalla per a cada secció de codi per així poder veure en quina zona del codi és queda aturat.
- En la **descàrrega d'imatges de flickr** s'ha trobat un altre problema. Flickr no té la imatge de més qualitat, per a totes les imatges que troba el cercador. Al membres del grup els interessa que les imatges que es descarreguin siguin de la més alta qualitat possible. Així doncs, la solució al problema és comprovar (abans de la descàrrega de la imatge) si l'enllaç on hauria de ser la imatge gran existeix o no. En cas que aquest enllaç no existeixi la imatge no es descarrega i es passa a la següent. Això provoca que el procés de baixada d'imatges sigui una mica més lent.
- Les **esperes en el llançament de mètodes**. En el moment d'executar un mètode el procés pot trigar segons o pot trigar molts minuts depenen de la quantitat d'imatges a processar. Això ens creava un problema. S'havia de fer saber al usuari que l'operació podria trigar una estona ja que si la web es quedava parada sense mostrar res molts dels usuaris podrien tancar pensant

que no funciona. La solució ha sigut crear una pàgina la qual s'obra des de la interfície web un moment abans d'executar el mètode. En aquesta pàgina es mostra una barra de procés que indica que es una operació llarga. Quan l'execució del mètode acaba desapareix i apareixen els resultats.

- **Execucions simultànies de mètodes** per diferents usuaris. Com que el pas del conjunt d'imatges, entre l'aplicació web i la interfície matlab es fa mitjançant arxius, aquests arxius han de ser únics.

Ens trobàvem amb el problema que quan dos o més usuaris consultaven al mateix temps els resultats es reescriuen en el mateix fitxer i per tan només un aconseguia la resposta correcta que donava la sortida del mètode executat.

Per a solucionar-ho es genera un nom d'arxiu únic per a cada consulta nova que es fa. Aquest nom surt del id de sessió. El que es fa es generar una nova sessió per a cada nova consulta.

- **La unió dels dos projectes** (el que s'està presentant i el de la ETSE). En la integració de les dues parts és on s'han tingut els problemes més difícils de trobar. Tots ells relacionats amb la depuració d'errors.

Ens trobàvem que la interfície matlab executava els mètodes però aquests no tornaven els resultats. El funcionament de cada projecte individualment era correcte, així doncs s'havia de buscar una solució per anar trobant els punts conflictius. La solució ha sigut crear un arxiu de *log* en el qual es van anotant les parts de mètode que van sent executades i així poder veure en quines zones es queda aturat. Un cop es trobava el punt conflictiu es soluciona l'error i es torna a executar. Així fins que la integració va ser correcte.

Un cop el funcionament de l'aplicació ha semblat correcte s'han fet unes proves per confirmar el funcionament. Per a la realització de les proves s'han guardat al servidor molts conjunts d'imatges ordenats per categories. S'ha aprofitat la funcionalitat de la web per a descarregar conjunts d'imatges de flickr per crear-ne alguns més (i així provar el funcionament d'aquesta part). Alguns d'aquests conjunts són els següents:

Textures : artificials, ceràmiques, edificis, núvols, menjars, metalls, papers, plantes , sorra , aigua, ...

Flickr : platges, ciutats, amb molts colors, deserts, prats verds, muntanyes nevades.

En total es tenen emmagatzemades unes 1800 imatges diferents.

Per fer la prova dels mètodes des de web s'han tingut que fer els càlculs previs d'aquestes 1800 imatges. Això s'ha fet localment i des de el matlab. Per fer proves s'ha pre calculat l'execució de dos dels tres mètodes que ja estan en funcionament.

Les comandes executades per a fer aquests càlculs previs són les següents:

➤ **Comanda per al mètode edison_wrapper**

```
imRetrieval('/home/group/projecte/projecte/file.txt','edison_wrapper','featurefun',@RGB2Luv,'steps',2,'synergistic',true,'spatialbandwidth',7,'rangebandwidth',6.5,'minimumregionarea',20,'speedup',2,'gradientwindowradius',2,'mixtureparameter',0.3,'edgestrengththreshold',0.3,'/home/group/projecte/output/fileOUT.txt','imatge','/home/group/uploadedImages/imatge.jpg','distancia','euclidean');
```

➤ **Comanda per al mètode histogram_intersection**

```
imRetrieval('/home/group/projecte/projecte/file.txt','histogram_intersection','num_bins',8,'/home/group/projecte/output/fileOUT.txt','imatge','/home/group/uploadedImages/imatge.jpg','distancia','euclidean');
```

En els dos casos el arxiu *file.txt* conté l'adreça de les 1800 imatges. Tots els processos de càlculs previs han tardat uns dos dies en total.

Un cop estan tots els càlculs previs estan realitzats és el moment de fer la prova del funcionament dels mètodes.

Primer s'han creat els arxius de configuració dels mètodes (*.method*) per a generar el seu formulari web. Un cop s'ha comprovat que el formulari es genera correctament és el moment d'executar-los amb una imatge de consulta.

Necessitem doncs, unes imatges de prova per a ser pujades al servidor i poder analitzar-les sobre tot el conjunt de 1800 aplicant un dels dos mètodes anteriors. Les imatges de prova seleccionades són les següents:



Figura 66. Imatges amb l'es que s'ha provat l'aplicació.

Amb la supervisió de l'expert en els mètodes i la visualització de resultats (tenint en compte la imatge que es consulta) es conclou que el funcionament de l'aplicació és correcte.

5. CONCLUSIONS

5.1. Valoració del projecte

Aquest projecte s'ha realitzat al llarg del curs 2008-2009. Després de la seva realització es pot fer una valoració de la feina feta amb relació amb el temps que s'hi ha dedicat.

El projecte ha constat de tres grans parts. Primer una fase en la que es van realitzar tots els estudis necessaris per a saber quines serien les tècniques i eines a utilitzar per aconseguir els objectius establerts. Un cop es van decidir les tècniques i eines a utilitzar, es va comprovar el seu funcionament (sempre amb proves locals per a cada eina o tècnica), s'ha passat a la següent fase.

Segon, el disseny de l'aplicació és un altre dels punts forts. És en aquest moment en el que s'han fet tots els mòduls utilitzant les eines i tècniques que s'havien estudiat i finalment s'han unit. La unió d'aquestes és una fase crítica en la que s'ha dedicat molt temps.

Finalment la integració de la web amb l'altre projecte que es realitzava a la ETSE ha sigut la part final del projecte. En aquest punt les reunions entre projectistes han estat superiors, ja que, les especificacions han anat canviant a mida que el projecte ha anat evolucionant. Això implicava estar al dia en els punts en els quals els dos projecte s'uneixen.

Com a valoració final del projecte, doncs, podem dir que s'han après nous coneixements com l'execució de mètodes matlab des de web, consulta i descàrrega d'imatges de flickr, muntatge d'un servidor web, creació d'una pàgina web amb un estil definit. També caldria remarcar que ens hem familiaritzat amb llenguatges que eren desconeguts o no s'hi havia treballat gaire com matlab, php , javascript, css i perl. Així doncs, la valoració de la realització del projecte és molt positiva, ja que, després de molt d'esforç s'ha aconseguit realitzar l'aplicació web que es demanava amb tots els requeriments necessaris. Ha estat un procés d'aprenentatge que ens ha permès adquirir experiència en quant a metodologia de treball i experiència en desenvolupar un projecte des de zero amb tot el que això comporta. A més ens deixa molt camins oberts pels quals continuar l'evolució d'aquest projecte.

5.2. Conclusions del projecte

Recordem els objectius més importants del projecte:

- **Divulgació científica mitjançant una web.**
- **Aplicació web flexible i portable.**
- **Coordinació amb un altre projecte de la ETSE.**

Aquests eren els tres principals objectius del treball. Podríem concloure doncs, que s'ha aconseguit crear una web en la qual, els investigadors del grup CAT del Centre de Visió per Computador de la Universitat Autònoma de Barcelona , podran afegir nous mètodes de forma molt flexible. Aquests mètodes són carregats automàticament per la web i són disponibles per a que qualsevol usuari d'internet els pugui executar per comprovar el seu funcionament. Això farà que la divulgació científica d'aquest grup pugui créixer , donant la possibilitat a la resta del mon, de conèixer quins són els seus estudis a partir de demostracions dels mètodes que han creat.

Com a funcionament intern, la web també permet als membres del grup crear-se les pròpies bases de dades d'imatges d'una manera molt interactiva i fàcil d'utilitzar. Això permetrà que el temps necessari per a crear aquestes bases de dades disminueixi. Gràcies a com s'ha configurat l'aplicació web, aquesta permet una gran portabilitat, el que significa que, canviant alguns paràmetres, fàcilment pot ser instal·lada per altres grups interessats. S'ha procurat que afegir un nou mètode es pugui realitzar d'una manera molt flexible i sense necessitat de tocar el codi de la web, el que implica que qualsevol persona sense coneixements amb programació de web sites pugui afegir el seus mètodes a la web sense l'ajut de cap expert.

La coordinació amb un altre projecte de la ETSE ens ha servit per comprovar que el funcionament de la web és correcte. És a dir, s'ha aconseguit que l'aplicació web tingui tres demostracions dels mètodes que aquest projectista a realitzat en el seu projecte.

Finalment, destacar que la web ja està penjada a Internet i en funcionament (<http://www.cat.uab.cat/projecte>) i que es poden provar les demostracions dels

mètodes comentats anteriorment. La web, a més, ha estat programada pensant amb possibles futures ampliacions el que significa que hi ha feina que no està en funcionament però està preparada per ser utilitzada (comentat al punt de disseny).

5.3. Treball futur

Els resultats obtinguts, si bé assoleixen tots els objectius proposats, poden ser ampliat en diferents aspectes per oferir una major entitat al projecte. Aquestes millores es descriuen a continuació:

- L'aplicació ha sigut programada des de navegadors mozilla , per tan, una millora important seria fer que pogués ser **visible des de qualsevol navegador**.
- L'aplicació de demostracions de mètodes permet afegir nous mètodes però per a característiques molt concretes (comentades al punt 3.2.3). Seria interessant , doncs, continuar millorant aquest aspecte per acabar aconseguint **una aplicació en que la majoria de mètodes en l'anàlisi del color i la textura puguin ser afegits** a la web per tindre la seva corresponent demostració.
- Per aconseguir el punt anterior serà necessari estudiar la possibilitat d'**afegir nous components per a la representació de mètodes en formularis web**, que necessitin camps especials que ara mateix no existeixen.
- En aquests moments l'aplicació web accepta mètodes amb un tipus de sortida concreta: conjunt d'imatges ordenades o una imatge gràfic amb un text associat. Seria un major grau de flexibilitat buscar **un sistema en el qual es pogués configurar el tipus de sortida esperada per a cada mètode**, per poder acabar tenint una aplicació en la qual qualsevol mètode investigat pugui ser afegit.
- **Creació d'un procés offline per executar mètodes**. Aquestes execucions offline es podrien fer des de el propi servidor i serien útils per a realitzar tots els processos de càlculs previs que els mètodes necessitin fer sobre els conjunts d'imatges consultats. Aquests processos no poden ser executats des de web ja que poden arribar a tardar dies. Per ara aquests càlculs previs són executats des de matlab. La idea seria executar els mètodes des de servidor i així en el

moment en que un usuari utilitzés la demo d'algun mètode obtindria els resultats molt més ràpidament.

- El **cercador d'imatges a flickr** té un petit problema. En alguns casos, no existeix la imatge trobada en la seva versió amb més resolució. Al grup de recerca els hi interessa treballar amb imatges amb resolució alta, per tan seria important **millorar** aquest aspecte **i fer que en el moment de la cerca només ens mostrés aquelles imatges de les que només existeix la imatge gran** sense que el procés de consulta sigui molt més lent. En el seu moment es va fer una prova i es va comprovar que el problema no era fer que el cercador mostrés les imatges que tenen imatge gran, si no que, estava en que el temps de consulta creixia molt.
- **Donar més opcions en la creació i visualització de mètodes** com per exemple, poder afegir una descripció explicativa per a cada camp del mètode o en el moment de veure resultats comprovar amb quines imatges s'havia fet la consulta.
- L'aplicació pot millorar també en la forma de visualització de les imatges tan en els resultats del mètodes , com en les consultes a flickr. Ens referim a millorar la part en que les imatges es mostren. Seria interessant **utilitzar la nova tecnologia web (Ajax)** per a que l'actualització de pàgines sigui més ràpida i que per exemple no ens tinguem que esperar a obtenir totes les imatges resultants de cop, i les puguem anar veient a mida que les anem obtenint. Milloraríem doncs, la interacció de l'usuari amb la web.
- Una altre **nova tècnica dels navegadors web** és un *plug-in* nou que ara incorporen. El seu nom és **colliris** i ofereix un forma eficaç d'observar imatges, preparat per als nous dispositius tàctils d'avui (ipods , ordinadors amb pantalla tàctil..). Oferir doncs als usuaris una altre manera de veure les imatges molt més còmode i efectiva per a trobar i/o examinar imatges.

BIBLIOGRAFIA

A les següents pàgines web és d'on s'ha tret tota la informació per a la realització del projecte descrit. Algunes de les referències bibliogràfiques ja estan descrites en el projecte.

<http://www.mathworks.com/> . Pàgina del Matlab.

<http://www.flickr.com/services/api/>. Informació de la API de flickr.

<http://es.wikipedia.org/wiki/PHP> . Informació de php.

<http://es.wikipedia.org/wiki/JavaScript> . Informació de javascript.

<http://es.wikipedia.org/wiki/Perl> . Informació de perl.

<http://es.wikipedia.org/wiki/css> . Informació de css.

<http://es.wikipedia.org/wiki/apache>. Informació del apache.

<http://www.apache.org/> . Pàgina del apache.

<http://www.forosdelweb.com/f18/> . Foros de consulta php.

<http://www.webestilo.com/foros/foro.phtml?foro=3>. Foros de consulta php.

<http://www.lawebdelprogramador.com/>. Foros de consulta php.

<http://www.forosdelweb.com/f53/> . Foros de consulta de css.